Anisotropic Solution Adaptive Unstructured Grid Generation Using AFLR

Sponsor: NASA – Langley Research Center

Grant No: NNL04AA91G

15 March 2007

PI: David L. Marcum

marcum@simcenter.msstate.edu

Mississippi State University

High-Performance Computing Collaboratory (HPC2)

Computational Simulation and Design Center (SimCenter)

2 Research Blvd.

Starkville, MS 39759

http://www.simcenter.msstate.edu

Anisotropic Solution Adaptive Unstructured Grid Generation Using AFLR

David L. Marcum

marcum@simcenter.msstate.edu

http://www.simcenter.msstate.edu

Mississippi State University
High-Performance Computing Collaboratory (HPC2)
Computational Simulation and Design Center (SimCenter)
2 Research Blvd.
Starkville, MS 39759

Abstract

An existing volume grid generation procedure, AFLR3, was successfully modified to generate anisotropic tetrahedral elements using a directional metric transformation defined at source nodes. The procedure can be coupled with a solver and an error estimator as part of an overall anisotropic solution adaptation methodology. It is suitable for use with an error estimator based on an adjoint, optimization, sensitivity derivative, or related approach. This offers many advantages, including more efficient point placement along with robust and efficient error estimation. It also serves as a framework for true grid optimization wherein error estimation and computational resources can be used as cost functions to determine the optimal point distribution. Within AFLR3 the metric transformation is implemented using a set of transformation vectors and associated aspect ratios. The modified overall procedure is presented along with details of the anisotropic transformation implementation. Multiple two- and three-dimensional examples are also presented that demonstrate the capability of the modified AFLR procedure to generate anisotropic elements using a set of source nodes with anisotropic transformation metrics. The example cases presented use moderate levels of anisotropy and result in usable element quality. Future testing with various flow solvers and methods for obtaining transformation metric information is needed to determine practical limits and evaluate the efficacy of the overall approach.

1. Introduction

Widespread access to high-performance-computing at reasonable costs combined with maturing CFD technology has led to extensive use of CFD simulations throughout industry. Within the aerospace community the complexity of configurations and physics being considered has steadily increased along with computational power. At present, one of the limiting issues in large-scale CFD simulations is the grid generation process and in particular optimization of the grid resolution throughout the field. This work explores the modification of an existing grid generation procedure, AFLR3, to address the grid optimization problem. Specifically, anisotropic solution adaptation was added to optimize the grid using directional metric transformations. The metric terms can be obtained from an adjoint or optimization based error estimation methodology. This approach offers many advantages, including more efficient point placement along with robust and efficient error estimation. It also serves as a framework for true grid optimization wherein error estimation and computational resources can be used as cost functions to determine the optimal point distribution.

AFLR3 (Advancing-Front/Local-Reconnection) is a state-of-the-art, well proven, and tested unstructured grid generator that is widely used throughout in the aerospace community for CFD simulations. AFLR3 uses a combination of automatic point creation, advancing-normal point placement, advancing-front point placement and connectivity optimization schemes to generate a high-quality tetrahedral/pentahedral grid. A valid grid is maintained throughout the AFLR grid generation process. This provides a framework for implementing efficient local search operations

using a simple data structure. It also provides a means for smoothly distributing the desired point spacing in the field using a point distribution function. This function is propagated through the field by interpolation from the boundary point spacing or by a specified growth normal to the boundaries. Points are generated using either advancing-front type placement for isotropic elements within the field or advancing-normal type point placement for surface orientated high-aspect ratio elements within boundary-layer regions. The connectivity for new points is initially obtained from direct subdivision and then optimized by iteratively using local-reconnection subject to a quality criterion. A combined Delaunay and min-max type (minimize the maximum angle) criterion is used. The overall procedure is applied repetitively until a complete field grid is obtained. High-quality isotropic and high-aspect ratio element two- and three-dimensional grids have been efficiently generated about geometrically complex configurations using this procedure.

AFLR3 currently includes an isotropic solution adaptation procedure that uses a set of adaptation sources derived from a user-defined error-estimate. For high-resolution adaptation available computing resources limit this mode of adaptation, as the overall grid size can quickly become prohibitive for large-scale simulations. An anisotropic solution adaptation/grid optimization scheme is needed to increase the level of resolution possible for a given computing resource budget.

AFLR3 does include some of the basis for an anisotropic adaptation/optimization scheme. For cases with high-aspect-ratio quad surface faces, e.g. near the leading edge of a wing, directional transformation metrics are used. These metrics are defined at the surface and propagated into the field to account for surface anisotropy and allow the local-reconnection procedure to work properly for Delaunay and other optimization criteria. This approach allows for very high-aspect-ratio quad surface faces and has been evaluated at ratios up to 10000:1. In the present work the field generation procedure was modified to allow for grid re-generation using directional metric transformations.

2. Anisotropic Transformation Implementation

The use of anisotropic transformation metrics within the field generation requires significantly more modification to the overall procedure than in the case of surface based metrics. Point placement, distance checking, interpolation, along with local-reconnection must all be modified to incorporate transformation metrics. Within AFLR3 the transformation metrics are defined by a set of transformation vectors and associated aspect ratios. Transformation properties are input to AFLR3 in the form of adaptation sources. For each source node, the coordinate location, desired isotropic point spacing, and transformation properties are specified. The source nodes can be directly inserted into the initial tessellation of the boundary surface points or they can be inserted into an oct-tree based background grid for interpolation within the field.

The overall grid generation procedure with modifications for anisotropic field grid generation is described below.

1) Specify point spacing on the boundary surface.
2) Generate a boundary surface grid.
3) Generate a valid initial tessellation of the boundary surface points only and recover all boundary surfaces. Use transformation properties defined by the aspect ratio of the surface grid faces to obtain the initial tessellation.
4) Assign a point distribution function to each boundary point based on the local point spacing.
5) Either insert the source nodes directly into the initial tessellation or create an oct-tree based source node background grid.
6) Initialize the transformation properties at boundary points.
7) For high-aspect-ratio boundary-layer elements, generate points using advancing-normal type point placement. Points are generated one layer at a time from the boundaries by advancing along normals dependent upon the boundary surface geometry.
8) For field elements outside the boundary-layer region, generate points using advancing-front type point placement with the "ideal" location defined in a local transformed space from the source node based transformation properties.
9) Interpolate the point distribution function and source node based transformation properties for new points from the containing elements and/or oct-tree based source node background grid.
10) Reject new points that are too close to other new points using distance checks in transformed spaced defined by the source node based transformation properties.
11) Insert the accepted new points by directly subdividing the elements that contain them.
12) Optimize the connectivity using local-reconnection. For each existing and possible element pair, compare the reconnection criterion for all reasonable configurations and reconnect using the most optimal one. Evaluate the quality criterion in transformed spaced defined by the source node based transformation properties. Repeat this local-reconnection process until no elements are reconnected. A combined Delaunay and min-max type criterion is used to improve the overall grid quality and overcome problems associated with optimum local states that are far from globally optimum.

13) Repeat the point generation and local-reconnection process, steps 8 through 12, until no new points are generated.

14) Smooth the coordinates of the field grid with the "ideal" location defined in transformed space from the source node based transformation properties.

15) Optimize the connectivity using the local-reconnection process (step 12).

Within AFLR3 anisotropic metric transformation is implemented using a local transformation that is defined by a set of orthogonal transformation vectors and corresponding aspect ratios. For two-dimensions, a circle in local transformed space will map to an ellipse in physical space. A set of transformation vectors ($\overline{T}_u$, $\overline{T}_v$) and corresponding aspect ratios (α_u, α_v) along with an associated physical space ellipse are illustrated in Fig. 1 for two-dimensional space.

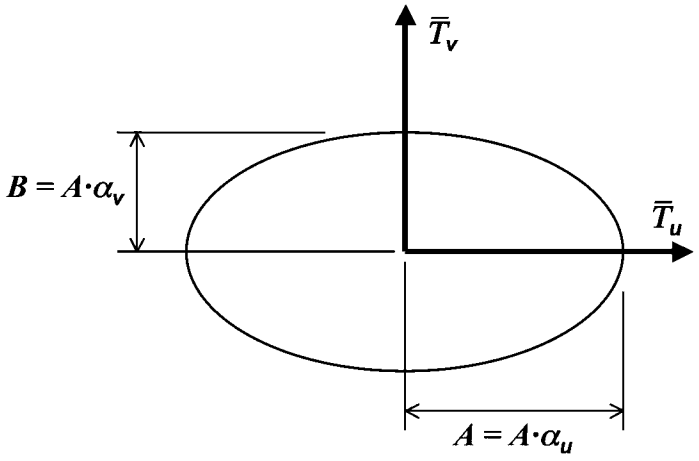

Fig. 1. Transformation ellipse and vectors for 2D.

For three-dimensions, a sphere in local transformed space will map to an ellipsoid in physical space. A set of transformation vectors ($\overline{T}_u$, $\overline{T}_v$, $\overline{T}_w$) and corresponding aspect ratios (α_u, α_v, α_w) along with an associated physical space ellipsoid are illustrated in Fig. 2 for three-dimensional space.

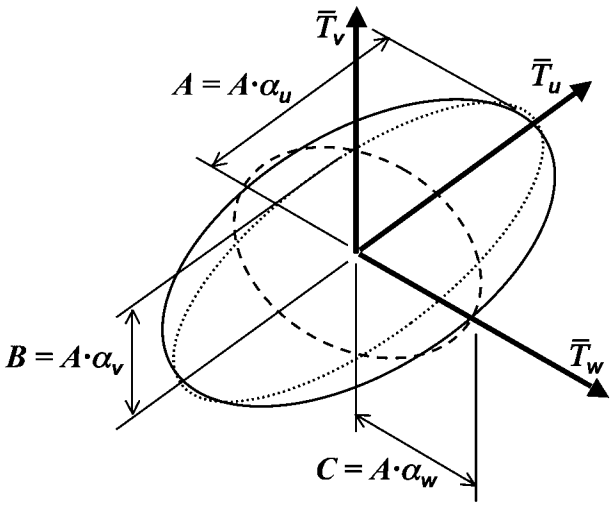

Fig. 2. Transformation ellipsoid and vectors for 3D.

The transformation vectors are defined as a set of orthogonal unit vectors given by the following.

$$|\bar{T}_u| = 1, |\bar{T}_v| = 1, |\bar{T}_w| = 1 \quad (1)$$

$$\bar{T}_u = \bar{T}_v \times \bar{T}_w \quad (2)$$

A corresponding set of aspect ratios (α_u, α_v, α_w) is used to determine the magnitude of the transformation in each direction. The aspect ratios are ratios of the ellipsoid semi-axes (***A, B, C***), given by

$$\alpha_u = (A / A) = 1 \quad (3)$$

$$\alpha_v = (A / B) \geq 1 \quad (4)$$

$$\alpha_w = (A / C) \geq 1 \quad (5)$$

As defined above, the ellipsoid semi-axis in the u-direction (***A***) is always the longest semi-axis. In two-dimensions, assuming the two-dimensional space is the x-y plane, the w-direction vector is defined simply as the unit vector in the z-direction and the w-direction aspect ratio (α_w) is one. The local magnitude of the semi-axes is dependent upon the local point spacing. Within AFLR, a distribution function (δf) is used to determine point spacing. It is equivalent to the isotropic point spacing and is initially determined from the point spacing of the input surface grid. As new field points are created within AFLR the distribution function is propagated by interpolation from existing points and/or source nodes (if any are specified as part of the input). With anisotropic transformation, the transformation vectors and corresponding aspect ratios are also propagated. As implemented, the distribution function (or isotropic spacing) should be interpreted as specifying the maximum local point spacing in any direction. And, the anisotropic transformation should be interpreted as specifying the minimum local spacing in particular directions. At any given point the actual length of the ellipsoid semi-axes (equivalent to spacing in each direction) can be obtained from the aspect ratios and the distribution function.

$$A = \alpha_u \cdot \delta f \quad (6)$$

$$B = \alpha_v \cdot \delta f \quad (7)$$

$$C = \alpha_w \cdot \delta f \quad (8)$$

For grid generation, anisotropic transformation need only be applied to displacement vectors for the difference in coordinates between points. Several grid generation related geometric operations involve a displacement vector ($\Delta\bar{X} = \bar{X}_i - \bar{X}_j$) between two points, e.g. distance between points, local-reconnection criterion, boundary normal vector, face area vector, and element volume. To account for anisotropic transformation the displacement vector is modified using the following transformation equations to obtain the final transformed displacement vector ($\Delta\bar{X}''$).

$$\Delta\bar{X}'' = \Delta\bar{X} + \bar{T}_v (\alpha_v - 1) (\bar{T}_v \bullet \Delta\bar{X}) + \bar{T}_w (\alpha_w - 1) (\bar{T}_w \bullet \Delta\bar{X}) \quad (9)$$

The transformation equation can be written as a sequence of equations for each transformation direction, if the transformation vectors are orthogonal.

$$\Delta\bar{X}' = \Delta\bar{X} + \bar{T}_v (\alpha_v - 1) (\bar{T}_v \bullet \Delta\bar{X}) \quad (10)$$

$$\Delta\bar{X}'' = \Delta\bar{X}' + \bar{T}_w (\alpha_w - 1) (\bar{T}_w \bullet \Delta\bar{X}') \quad (11)$$

Averaging or interpolation is required to obtain the transformation vectors and aspect ratios at the appropriate location for the transformation equations above. Interpolation is also required to evaluate those quantities at the location of new points. If sources are used then these quantities and the distribution function are interpolated from nearby sources. Alternatively, or without sources, they are interpolated from the existing element that contains the location.

For cases with sources, the nearby sources are determined from an oct-tree search. Weighted averages of the quantities at nearby sources are then determined. Given a location ($\bar{X}_i$) and a nearby source location ($\bar{X}_{src}$), the displacement vector ($\Delta\bar{X}_{i,src}$) and distance between them ($\Delta s_{i,src}$) are given by the following.

$$\Delta\bar{X}_{i,src} = \left|\bar{X}_i - \bar{X}_{src}\right| \tag{12}$$

$$\Delta s_{i,src} = \left|\Delta\bar{X}_{i,src}\right| \tag{13}$$

The distribution function associated with a given source can then be obtained using geometric growth from the source by

$$\delta f_{i,src} = \delta f_{src} + (\mu \cdot \Delta s_{m,i,src} - \Delta s_{m,i,src}) \tag{14}$$

where μ is the geometric growth rate and $\Delta s_{m,i,src}$ is a limited distance from the source location. The distance is limited by a no-growth exclusion zone defined as

$$\Delta s_m = max\ (\Delta s_{i,src} - \gamma \cdot \delta f_{src},\ 0) \tag{15}$$

where γ is the exclusion zone factor. Note that within AFLR the geometric growth rate and exclusion zone factor for sources are equivalent to the parameters ***cdfrsrc*** and ***cdfssrc***. The transformation aspect ratios can be obtained in a manner similar to the distribution function.

$$\alpha_{v,i,src} = max\left(\alpha_{v,src} / [1 + \alpha_{v,src}\ (\delta f_{i,src} - \delta f_{src}) / \delta f_{src}],\ 1\right) \tag{16}$$

$$\alpha_{w,i,src} = max\left(\alpha_{w,src} / [1 + \alpha_{w,src}\ (\delta f_{i,src} - \delta f_{src}) / \delta f_{src}],\ 1\right) \tag{17}$$

The transformation vectors associated with a source are simply the values defined at the source.

$$\bar{T}_{v,i,src} = \bar{T}_{v,src} \tag{18}$$

$$\bar{T}_{w,i,src} = \bar{T}_{w,src} \tag{19}$$

All of the quantities associated with the given source are weighted by the inverse of the distance from the source squared and then summed for all sources nearby the given location. The resulting set of summation equations for sources are listed below.

$$\omega_{i,src} = \Delta s_{i,src}^{-2} \tag{20}$$

$$\delta f_{i,src} = \sum(\omega_{i,src} \cdot \delta f_{i,src}) / \sum\omega_{i,src} \tag{21}$$

$$\alpha_{v,i,src} = \sum(\omega_{i,src} \cdot \alpha_{v,i,src}) / \sum\omega_{i,src} \tag{22}$$

$$\alpha_{w,i,src} = \sum(\omega_{i,src} \cdot \alpha_{w,i,src}) / \sum\omega_{i,src} \tag{23}$$

$$\bar{T}_{v,i,src} = \sum(\sigma_{v,i,src} \cdot \omega_{i,src} \cdot \bar{T}_{v,i,src}) / \left|\sum(\sigma_{v,i,src} \cdot \omega_{i,src} \cdot \bar{T}_{v,i,src})\right| \tag{24}$$

$$\bar{T}_{w,i,src} = \sum(\sigma_{w,i,src} \cdot \omega_{i,src} \cdot \bar{T}_{w,i,src}) / \left|\sum(\sigma_{w,i,src} \cdot \omega_{i,src} \cdot \bar{T}_{w,i,src})\right| \tag{25}$$

In the above equations, $\omega_{i,src}$ is the weight associated with each source, $\sigma_{v,i,src}$ and $\sigma_{w,i,src}$ are sign functions, and $\sum$ is the sum over all nearby sources. Note that the *sign* function is used to keep all local transformation vectors orientated in similar directions for summation. The overall sign for the transformation vectors themselves has no effect in the transformation equations, Equations (9) through (11). The displacement vector is chosen as the test vector for the sign functions given below.

$$\sigma_{v,i,src} = sign(\Delta \bar{X}_{i,src} \bullet \bar{T}_{v,i,src}) \tag{26}$$

$$\sigma_{w,i,src} = sign(\Delta \bar{X}_{i,src} \bullet \bar{T}_{w,i,src}) \tag{27}$$

For interpolation from existing points, summation equations similar to those for sources, Equations (20) through (27), can be used. The weight associated with each existing point is replaced with that for linear interpolation. Given a location ($\bar{X}_i$) and a subset of nearby existing points ($\bar{X}_j$) the interpolated quantities can be found using the following equations.

$$\sum \omega_{i,j} = 1 \tag{28}$$

$$\delta f_{i,j} = \sum(\omega_{i,j} \cdot \delta f_j) \tag{29}$$

$$\alpha_{v,i,j} = \sum(\omega_{i,j} \cdot \alpha_{v,j}) \tag{30}$$

$$\alpha_{w,i,j} = \sum(\omega_{i,j} \cdot \alpha_{w,j}) \tag{31}$$

$$\bar{T}_{v,i,j} = \sum(\sigma_{v,j} \cdot \omega_{i,j} \cdot \bar{T}_{v,j}) / \left|\sum(\sigma_{v,j} \cdot \omega_{i,j} \cdot \bar{T}_{v,j})\right| \tag{32}$$

$$\bar{T}_{w,i,j} = \sum(\sigma_{w,j} \cdot \omega_{i,j} \cdot \bar{T}_{w,j}) / \left|\sum(\sigma_{w,j} \cdot \omega_{i,j} \cdot \bar{T}_{w,j})\right| \tag{33}$$

In the above equations, $\omega_{i,j}$ is the linear interpolation weight associated with each existing point, $\sigma_{v,i,j}$ and $\sigma_{w,i,j}$ are sign functions, and $\sum$ is the sum over the subset of existing points used for interpolation. The subset of existing points is typically the points for an edge, face or element. The appropriate direction to align the transformation vectors is determined using the sign functions given below.

$$\sigma_{v,j} = sign(\bar{\nu}_v \bullet \bar{T}_{v,j}) \tag{34}$$

$$\sigma_{w,j} = sign(\bar{\nu}_w \bullet \bar{T}_{w,j}) \tag{35}$$

In the above equations, $\bar{\nu}_v$ and $\bar{\nu}_w$ are the test vectors. The test vectors ($\bar{\nu}_v$, $\bar{\nu}_w$) are chosen to be the transformation vectors at the existing point with the maximum v-direction aspect ratio.

$$jmax = j \text{ where } \alpha_{v,j} = max\,(\alpha_{v,j}) \tag{36}$$

$$\bar{\nu}_v = \bar{T}_{v,jmax} \tag{37}$$

$$\bar{\nu}_w = \bar{T}_{w,jmax} \tag{38}$$

The interpolation equations for the transformation quantities given by Equations (28) through (38) are appropriate for situations where the given location ($\bar{X}_i$) is well defined. For operations where the given location is a region, such as an element, use of values at a selected existing point can be more appropriate. Typical example operations include checking distance between points, evaluating a quality metric for a given element, or evaluating local-reconnection criteria for a group of elements. In such cases, the transformation vectors and associated aspect ratios can be set to the values at the existing point of the subset where the aspect ratio in the v-direction is at the maximum value. The subset of existing points searched is those points associated with the

region, such as the points of an element or group of elements. For a subset of existing points the maximum value location (***jmax***) is given by Equation (36) and the transformation properties are given by the following.

$$\alpha_{v,i,j} = \alpha_{v,jmax} \tag{39}$$

$$\alpha_{w,i,j} = \alpha_{w,jmax} \tag{40}$$

$$\bar{T}_{v,i,j} = \bar{T}_{v,jmax} \tag{41}$$

$$\bar{T}_{w,i,j} = T_{w,jmax} \tag{42}$$

3. Anisotropic Examples

Multiple two- and three-dimensional examples are presented in this section that demonstrate the capability of the modified AFLR procedure to generate anisotropic elements using a set of source nodes with anisotropic transformation metrics. The example cases presented here use moderate levels of anisotropy of about 3:1. Overall element quality, evaluated in physical space, for these cases is in general usable, however it is inferior to that for similar isotropic cases. The procedure has been successfully tested with much higher levels of anisotropy, although element quality, evaluated in physical space, degrades very quickly. In transformed space the element quality is much better and if that space is similar to solution space then solver performance may not suffer. Future testing with various flow solvers and methods for obtaining transformation metric information is needed to determine practical limits and evaluate the efficacy of the overall approach.

Two-dimensional examples are presented to demonstrate the upper-end of element alignment and quality. Figs. 3 through 6 present results for a set of source nodes in a cross pattern enclosed within a box. And, in Figs. 7 through 10 similar results are presented for a set of source nodes in a wave pattern enclosed within a box. All two-dimensional cases presented use growth in the transformation aspect ratio. Two modes of operation are presented; direct insertion of the source nodes into the initial triangulation (Figs. 3, 4, 7 and 8) and evaluation of the transformation properties and point spacing from a background grid of the same source nodes (Figs. 5, 6, 9 and 10). These cases all result in very good element quality in both transformed and physical space. They also result in a good alignment of the elements near the source locations. While, alignment carries over to some extent in three-dimensions, similar levels are not possible in general and alignment becomes more difficult to visualize.

Three-dimensional examples are presented in the remaining figures. All of the three-dimensional test cases use a pattern of source nodes within a cube. Within AFLR3, anisotropic source nodes can be input directly from a VNODE file (*.vnode) or they can be derived from an embedded surface. Note that running “aflr3 –help” will list more information on VNODE files and options related to anisotropic sources. In the case of an embedded surface the aspect ratio is determined from the ratio of the local point spacing divided by the initial normal spacing specified in the input boundary surface grid file. Figs. 11 through 19 present results for a plate pattern of sources, Figs. 20 through 28 present results for a spherical pattern, Figs. 29 through 31 present results for a spherical and x-wave pattern combined, Figs. 32 through 36 present results for a wave pattern, Figs. 37 through 41 present results for an x-plate pattern, and finally Figs. 42 through 46 present results for an x-wave pattern.

Comparative results are presented for some of the cases using isotropic sources (with point spacing reduced to match the smallest anisotropic point spacing). Obviously, considerably more points and elements are required to achieve the anisotropic point spacing level. Cases are also presented with and without anisotropic aspect ratio growth (AFLR3 options mtr=1 and mtr=2 respectively). Overall, it appears that the use of anisotropic growth (mtr=2) would produce more useful results in most cases. In addition, direct insertion of the source nodes into the initial tessellation is compared to use of a background grid (AFLR3 options msource=1 and msource=2). Results are similar between direct insertion and a background grid. If the source nodes are well distributed then direct insertion is more efficient and results in maximum alignment of the elements.

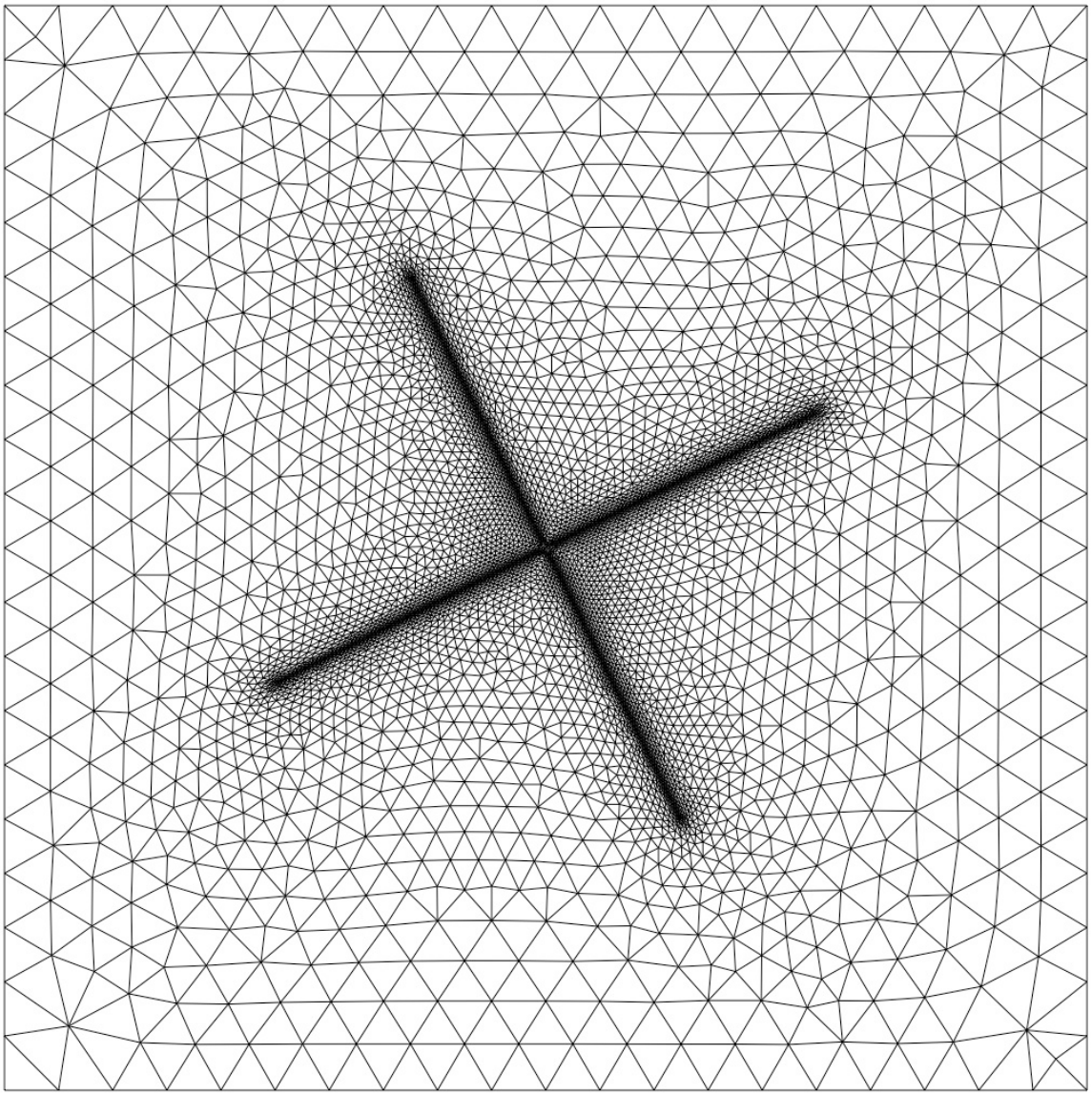

Fig. 3. 2D box-cross grid for anisotropic case with direct source insertion and with transformation vector growth.

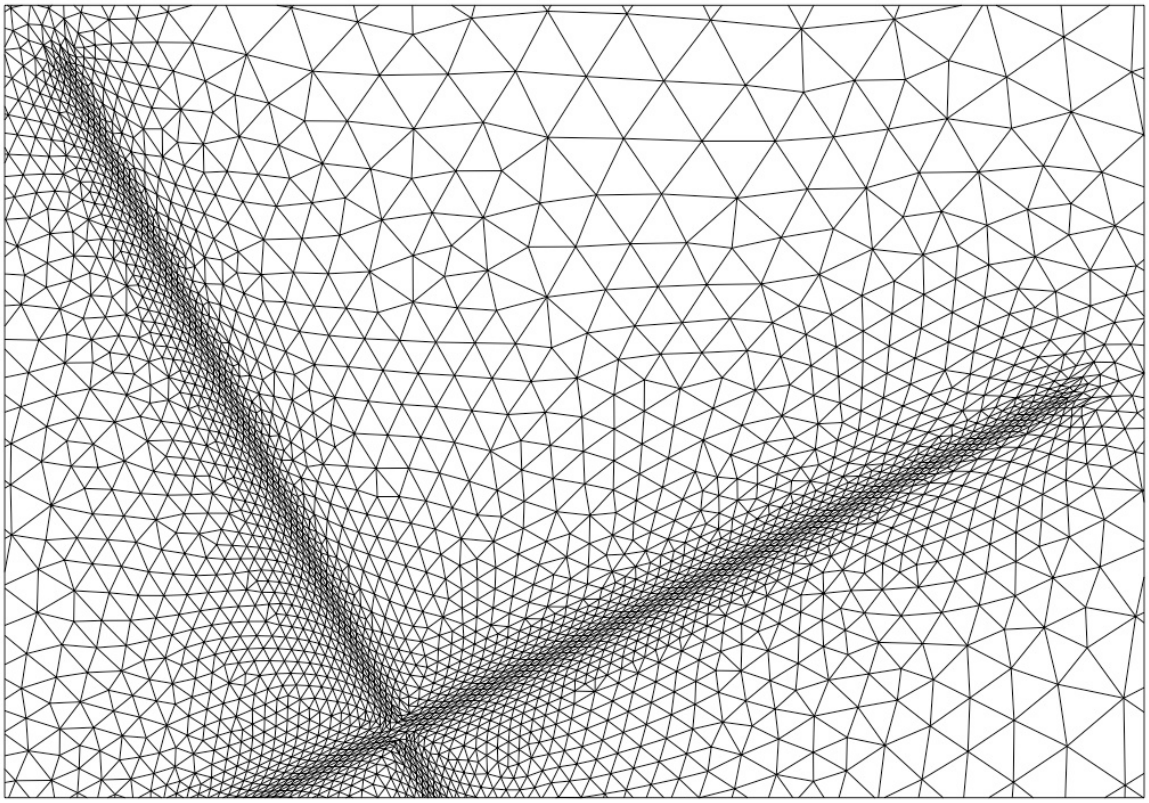

Fig. 4. Detail view of 2D box-cross grid for anisotropic case with direct source insertion and with transformation vector growth.

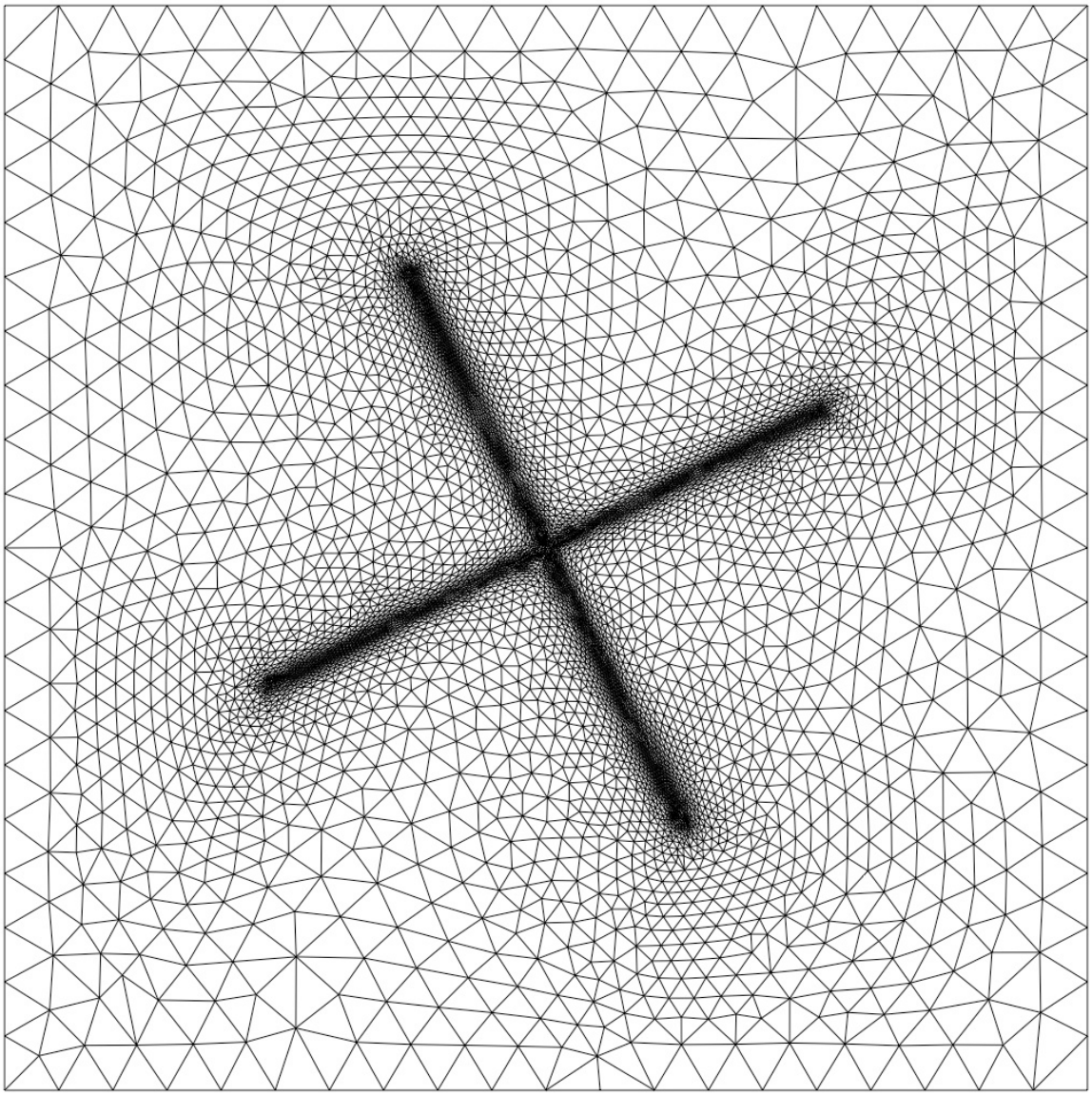

Fig. 5. 2D box-cross grid for anisotropic case with background source grid and with transformation vector growth.

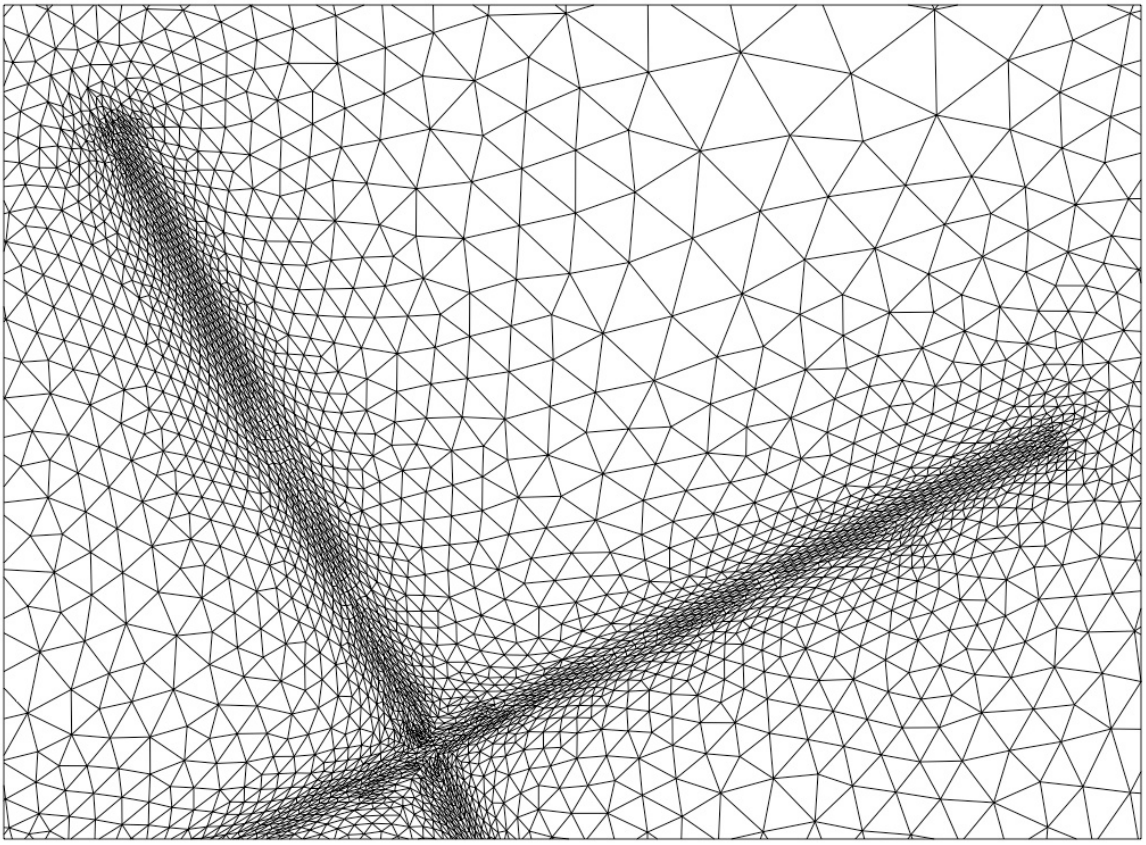

Fig. 6. Detail view of 2D box-cross grid for anisotropic case with background source grid and with transformation vector growth.

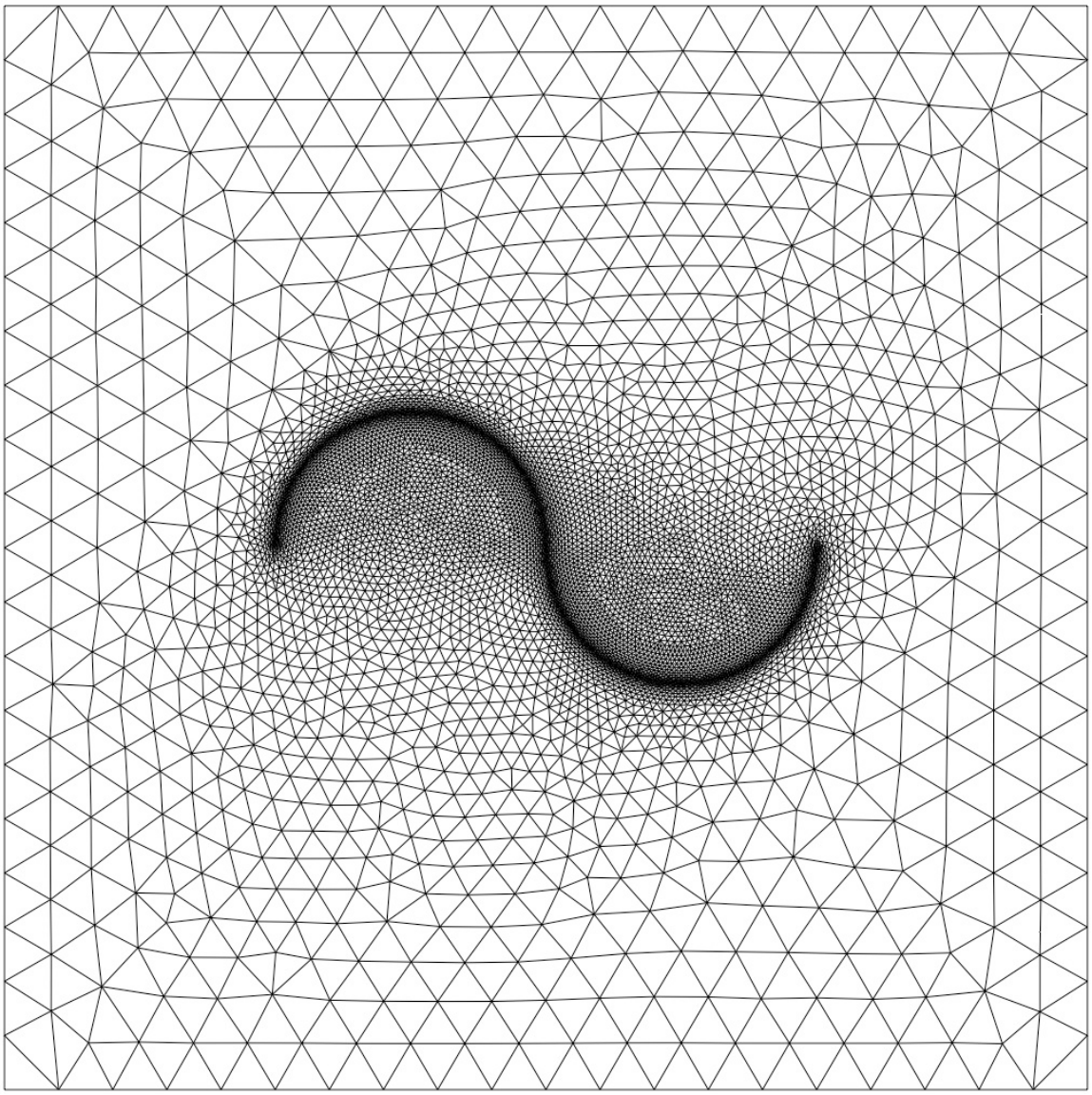

Fig. 7. 2D box-wave grid for anisotropic case with direct source insertion and with transformation vector growth.

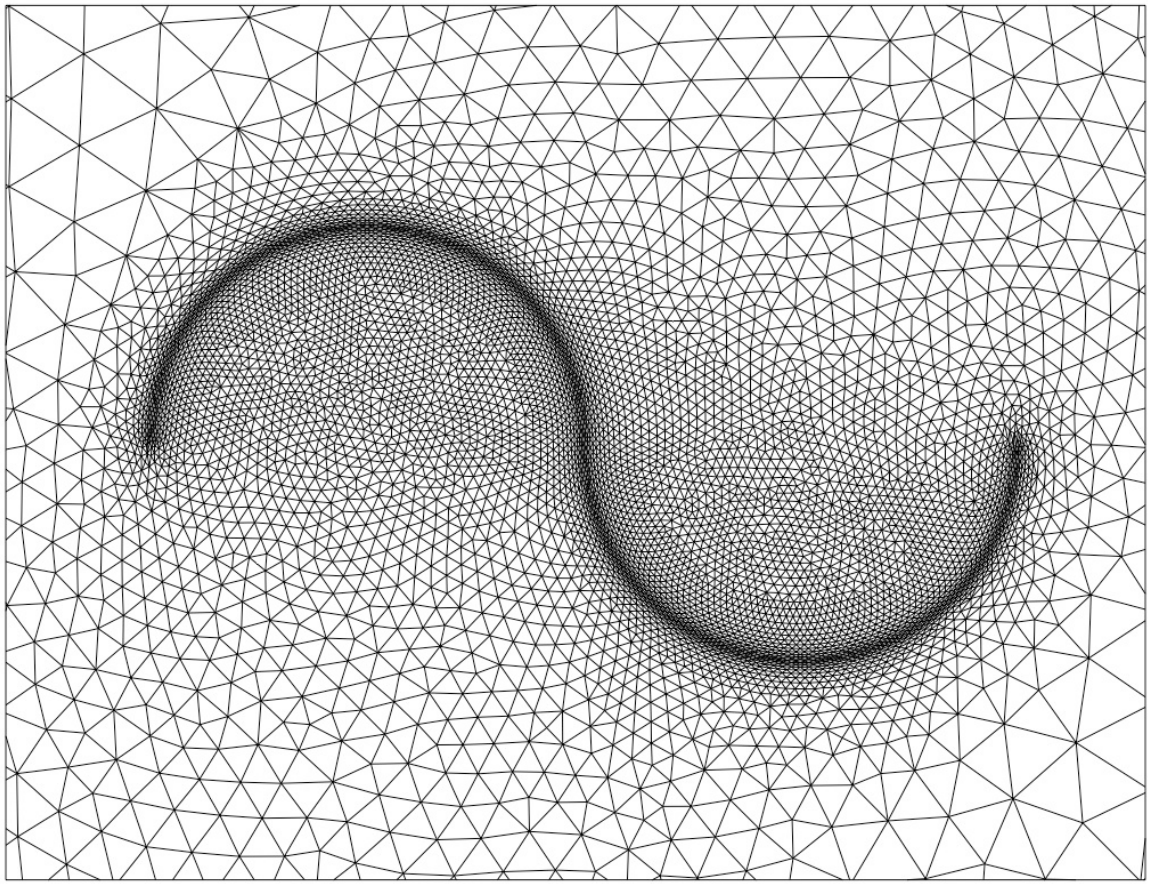

Fig. 8. Detail view of 2D box-wave grid for anisotropic case with direct source insertion and with transformation vector growth.

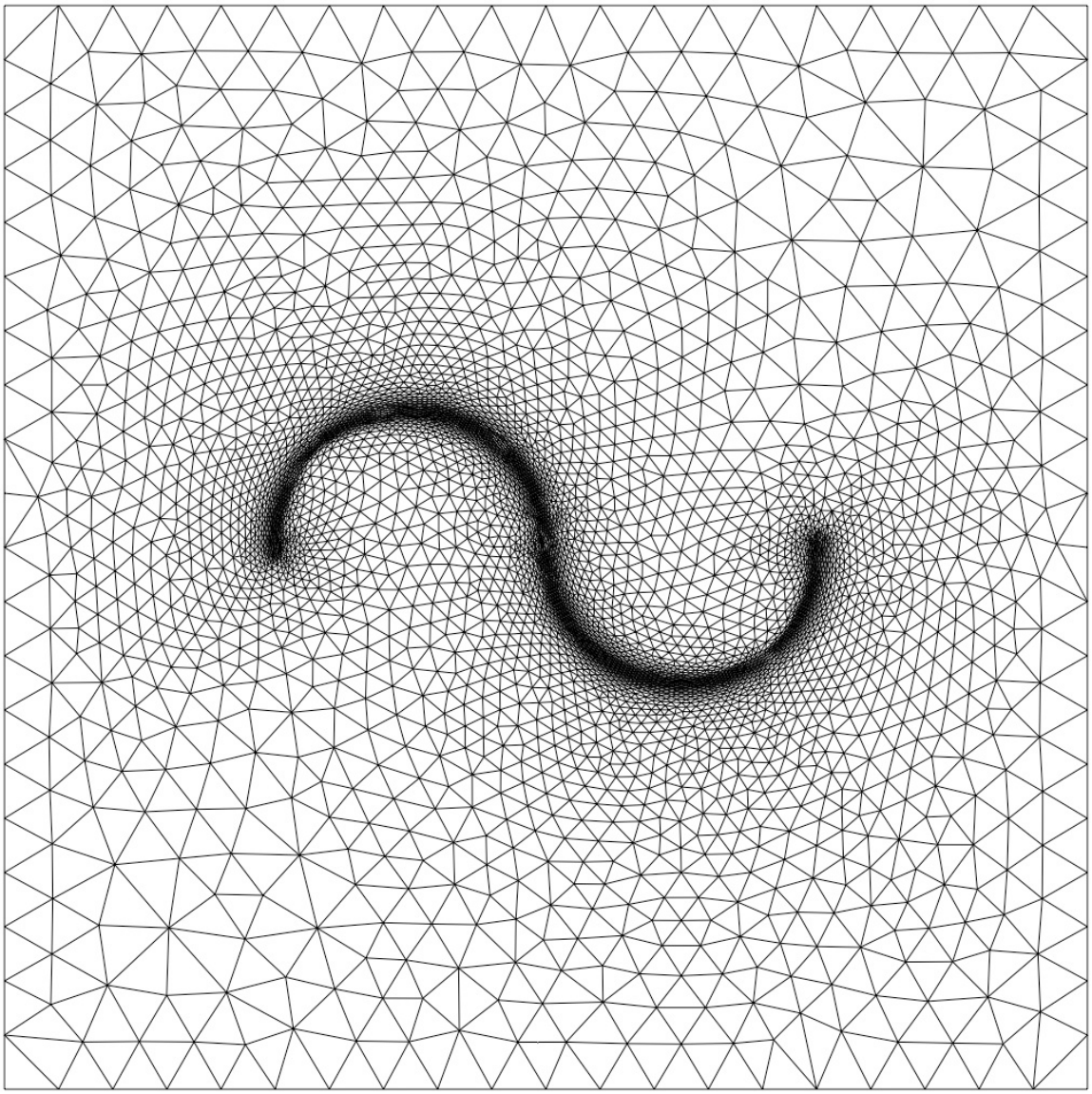

Fig. 9. 2D box-wave grid for anisotropic case with background source grid and with transformation vector growth.

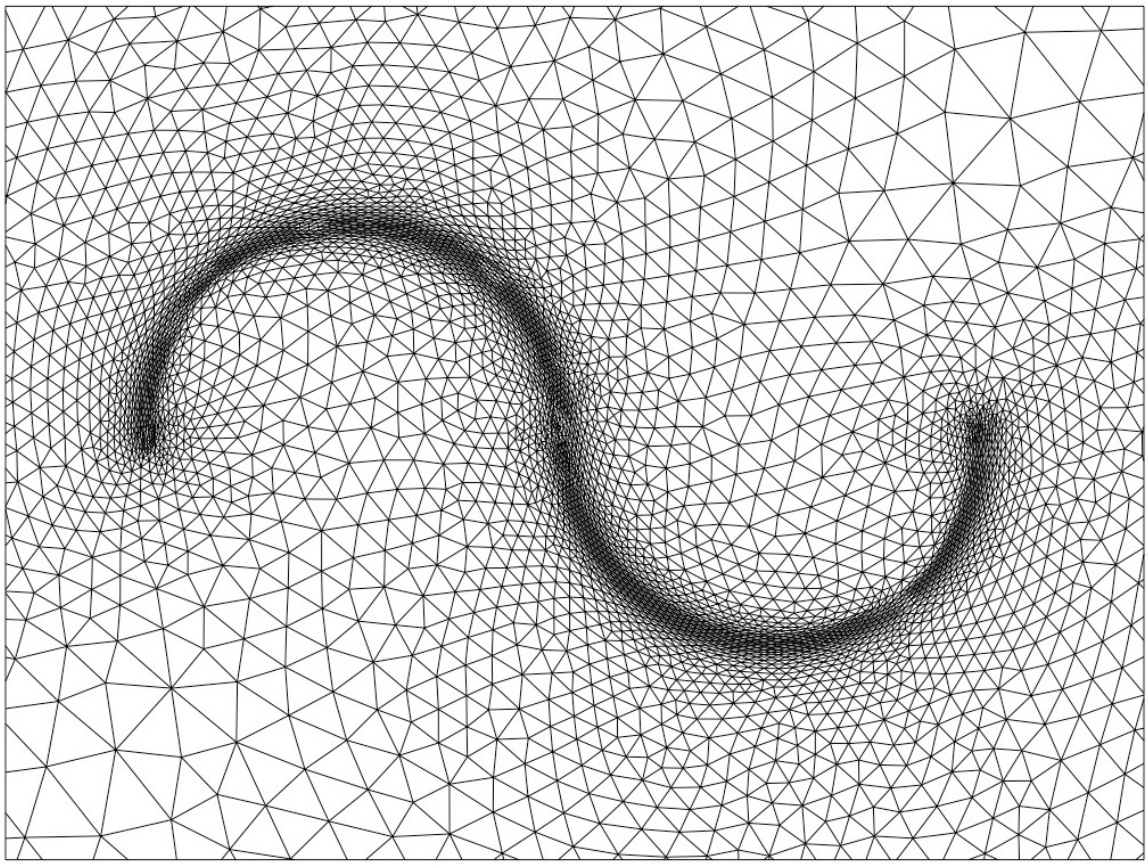

Fig. 10. Detail view of 2D box-wave grid for anisotropic case with background source grid and with transformation vector growth.

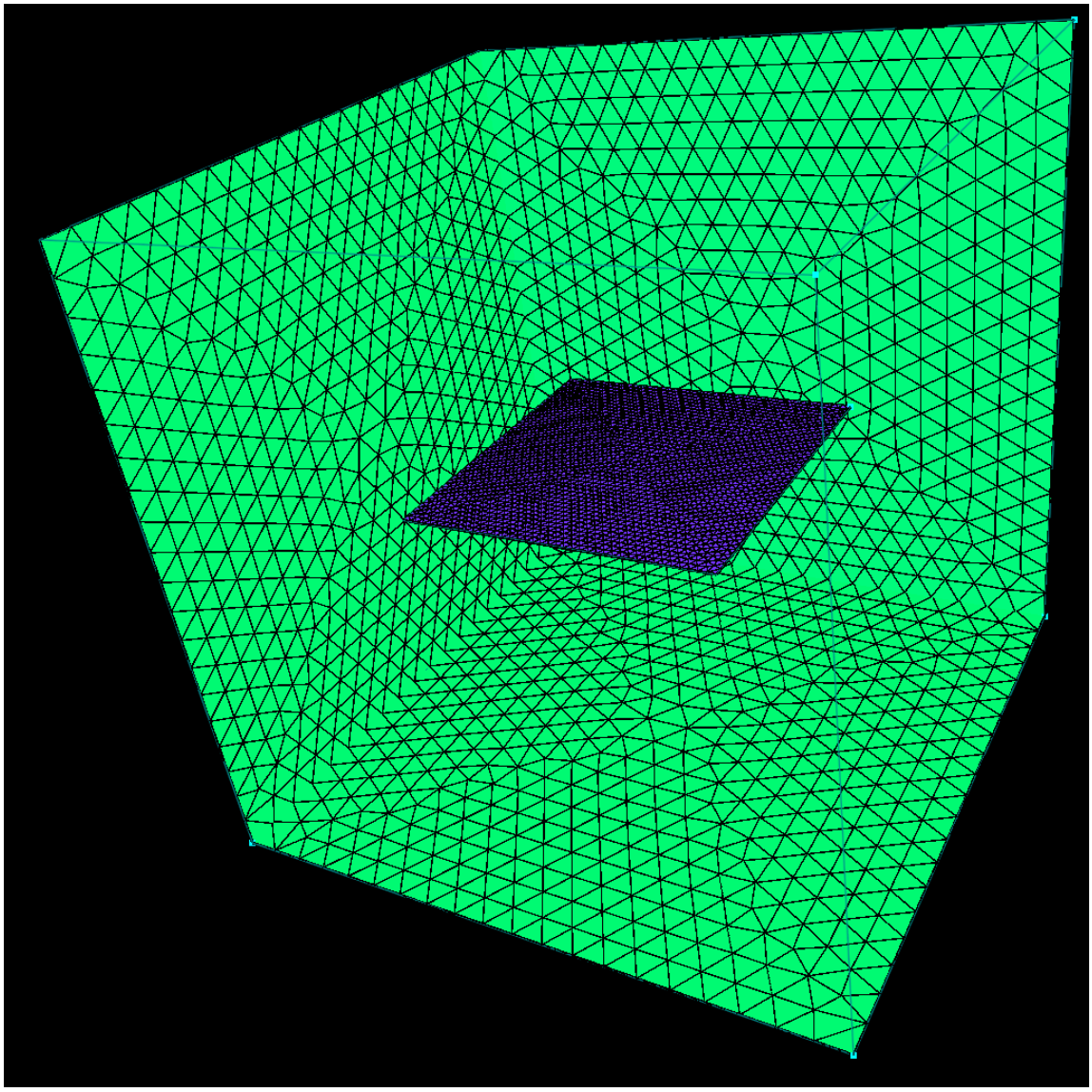

Fig. 11. Plate case surface grid.

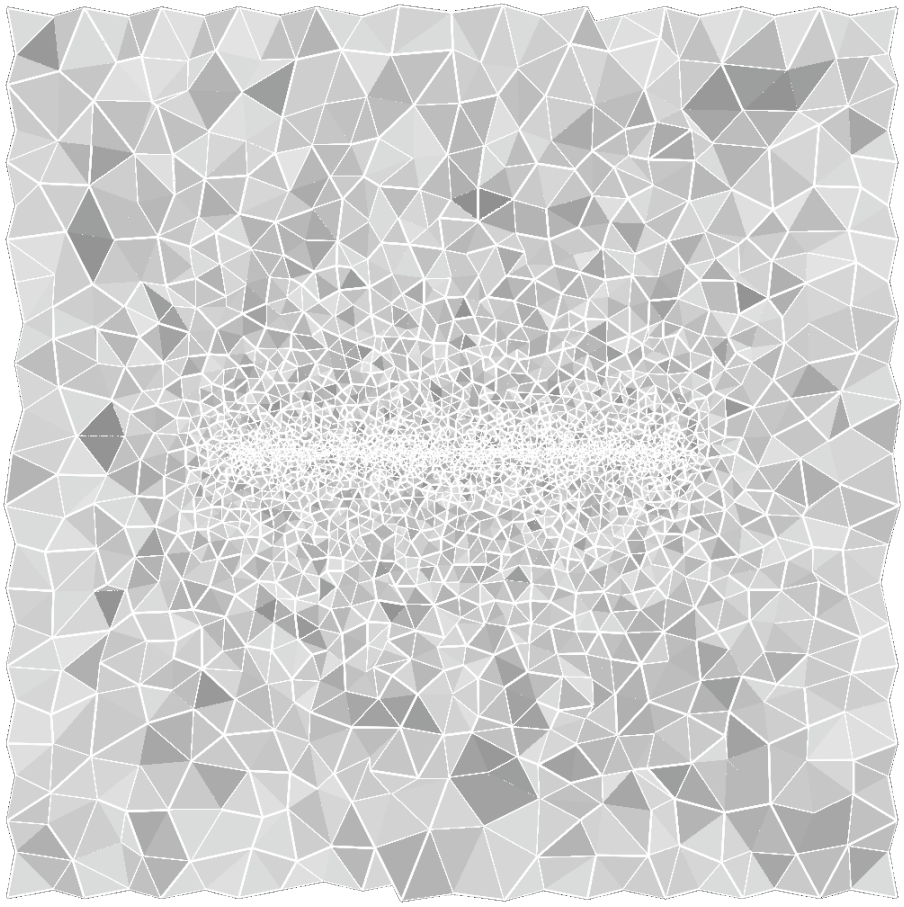

Fig. 12. Plate volume grid for isotropic case with direct source insertion.

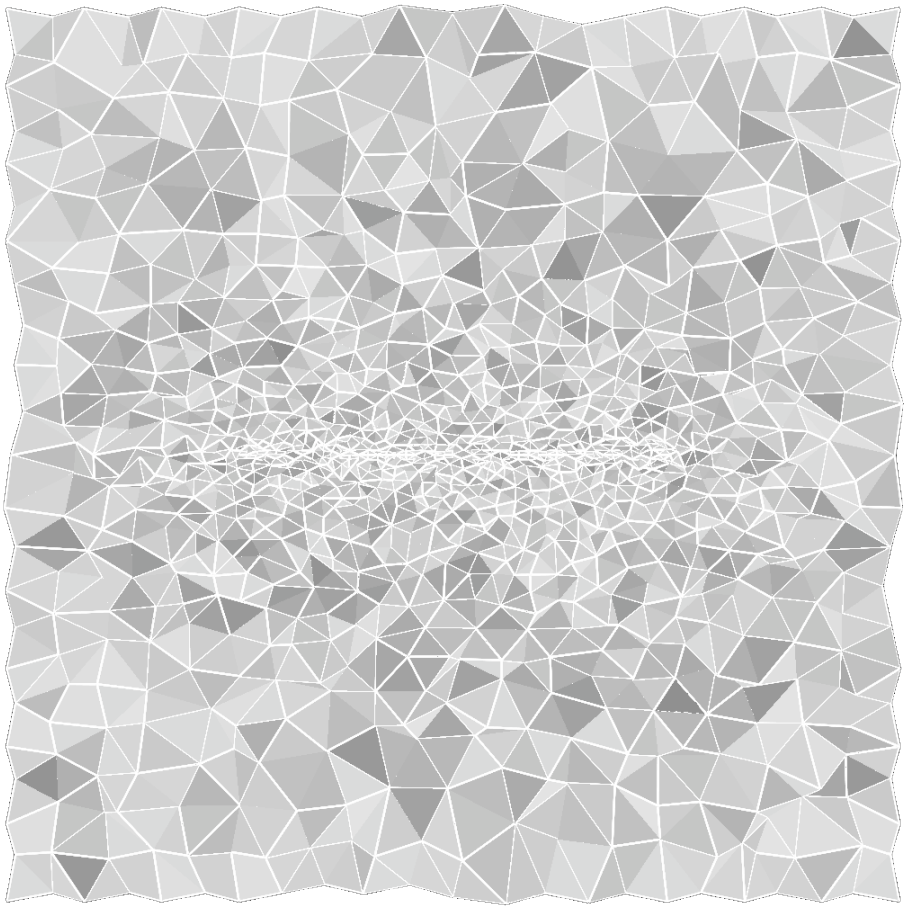

Fig. 13. Plate volume grid for anisotropic case with direct source insertion and without transformation vector growth.

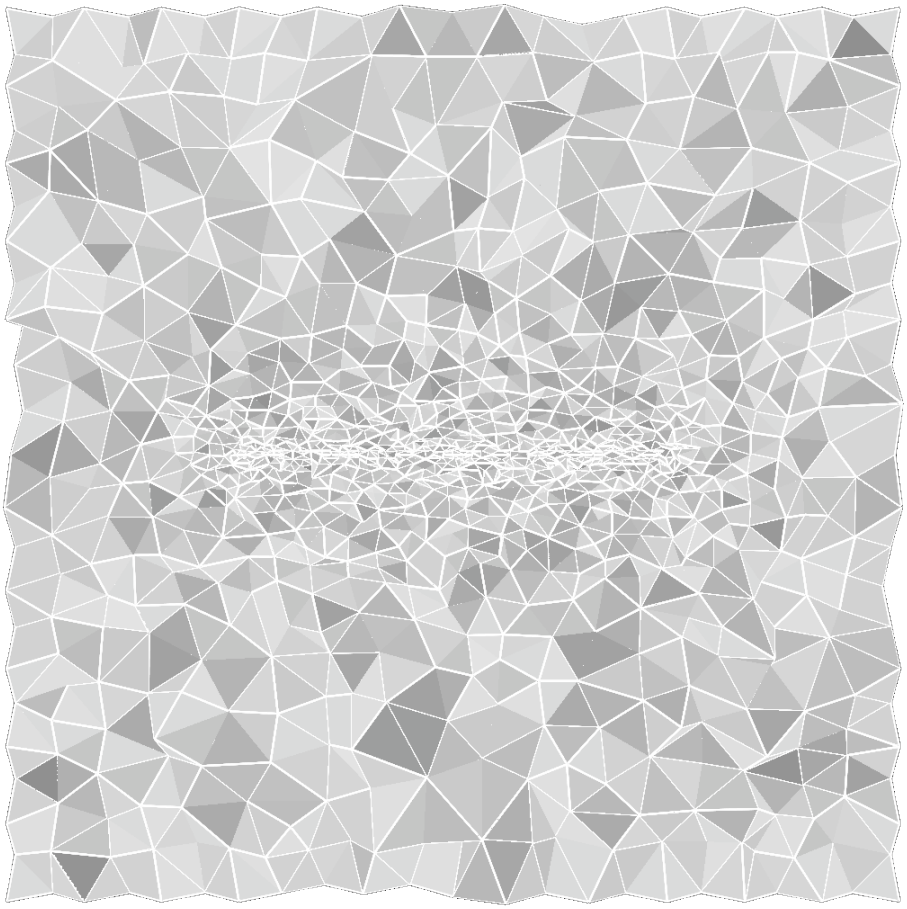

Fig. 14. Plate volume grid for anisotropic case with direct source insertion and with transformation vector growth.

Fig. 15. Detail view of plate volume grid for anisotropic case with direct source insertion and with transformation vector growth.

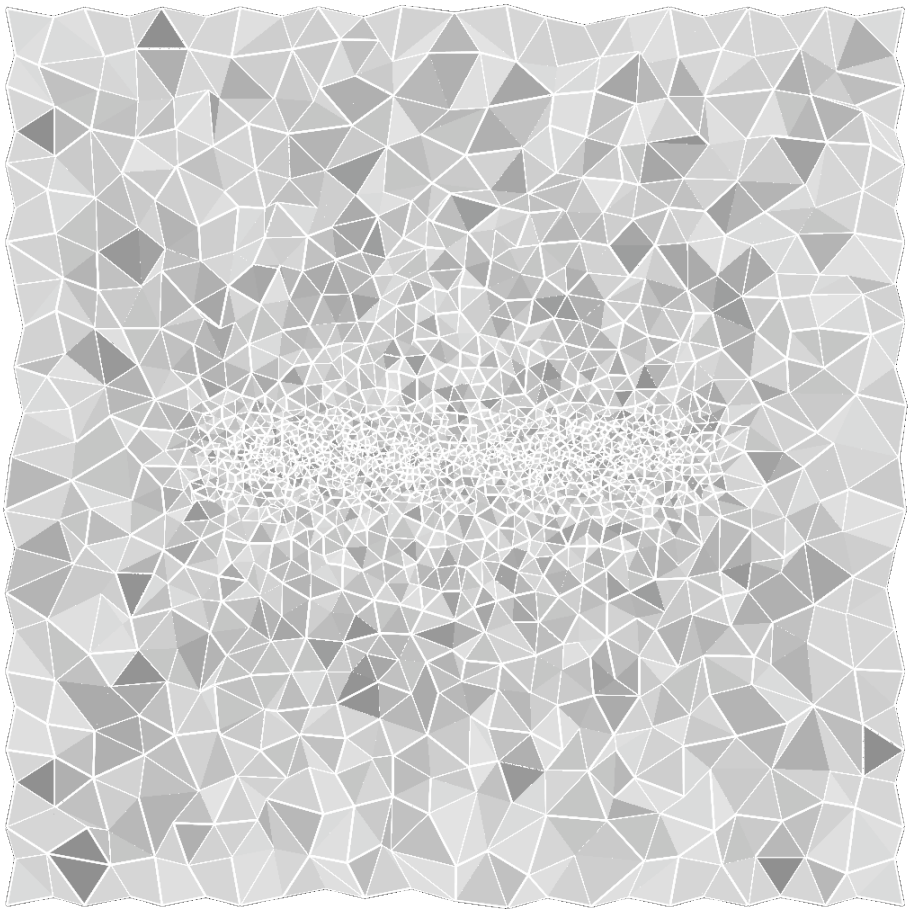

Fig. 16. Plate volume grid for isotropic case with background source grid.

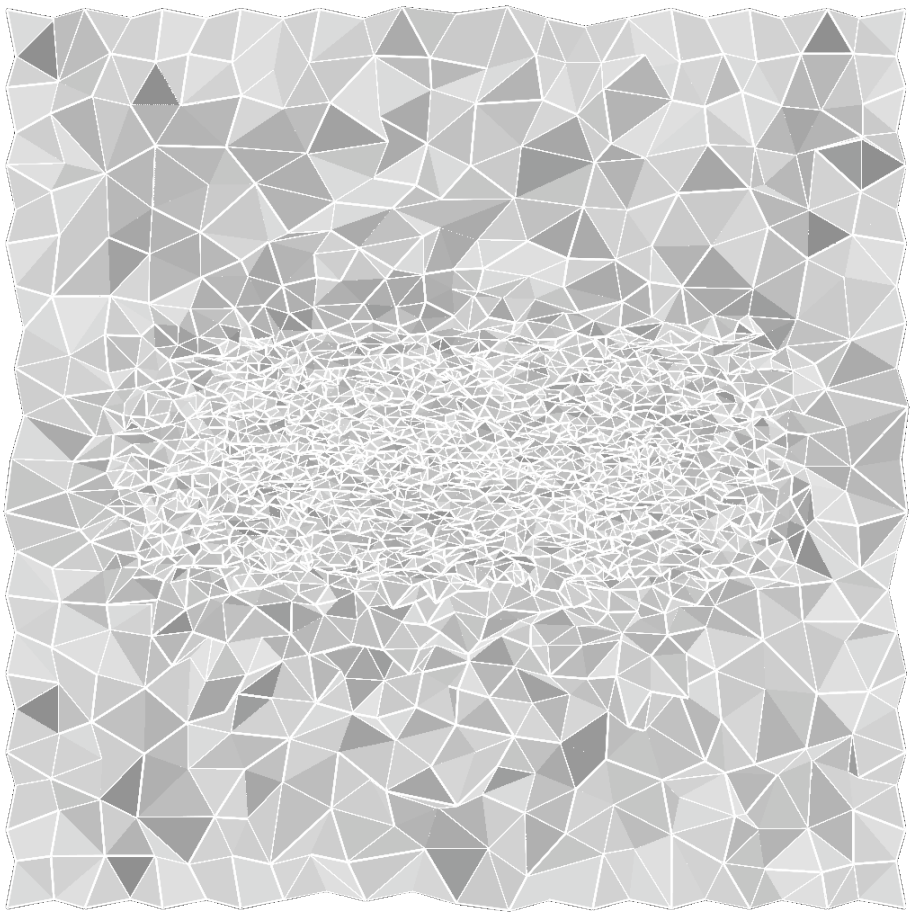

Fig. 17. Plate volume grid for anisotropic case with background source grid and without transformation vector growth.

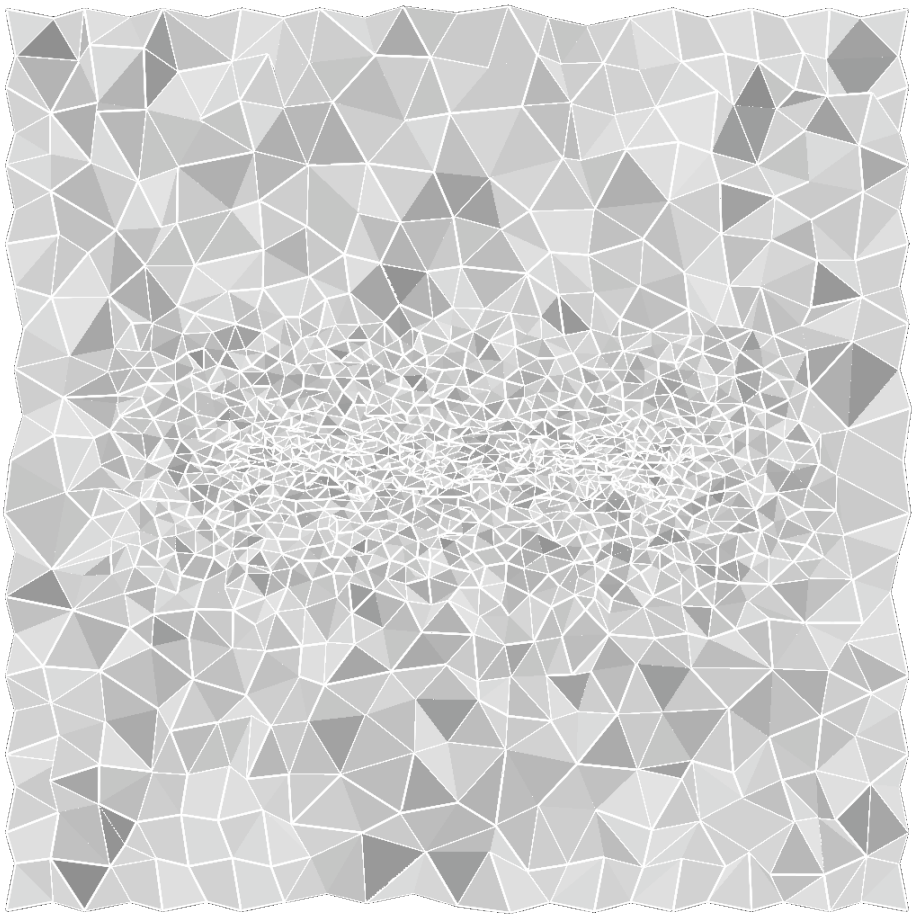

Fig. 18. Plate volume grid for anisotropic case with background source grid and with transformation vector growth.

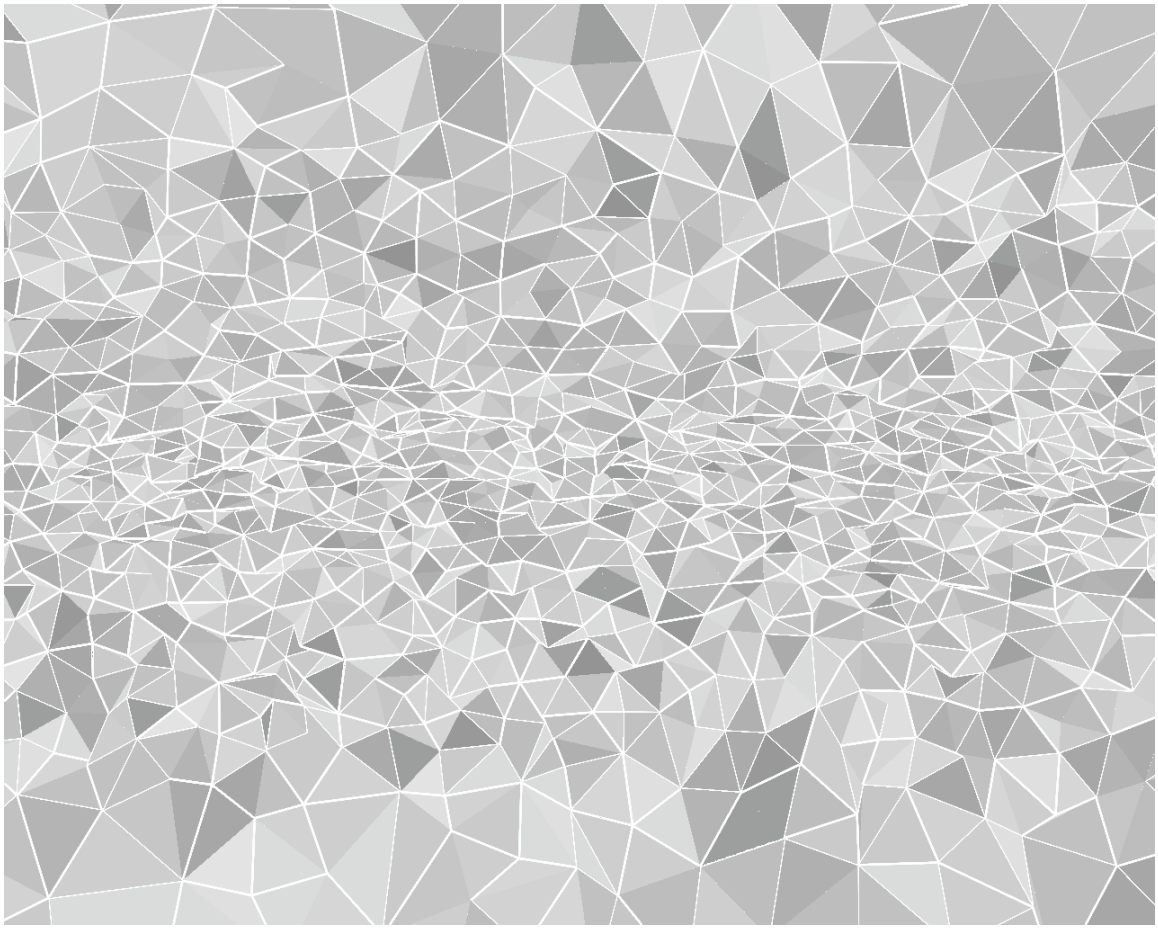

Fig. 19. Detail view of plate volume grid for anisotropic case with background source grid and with transformation vector growth.

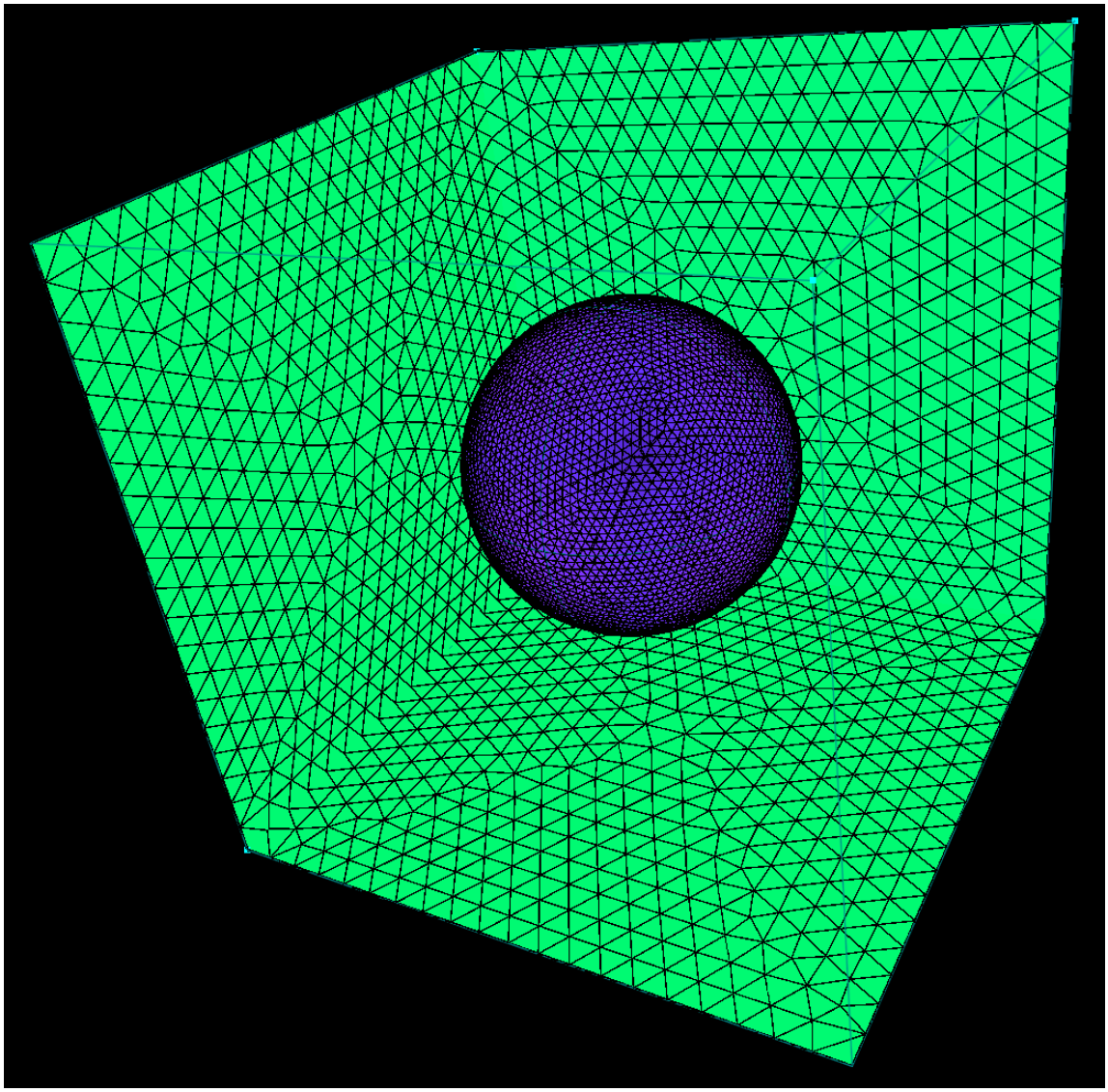

Fig. 20. Sphere case surface grid.

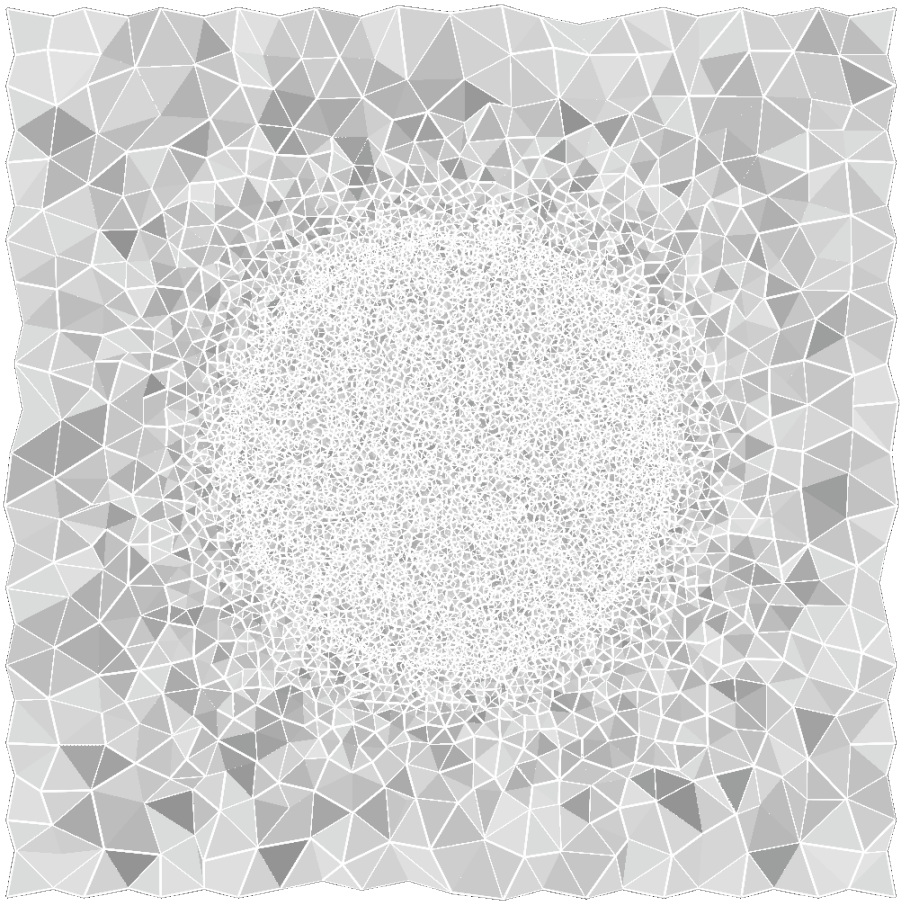

Fig. 21. Sphere volume grid for isotropic case with direct source insertion.

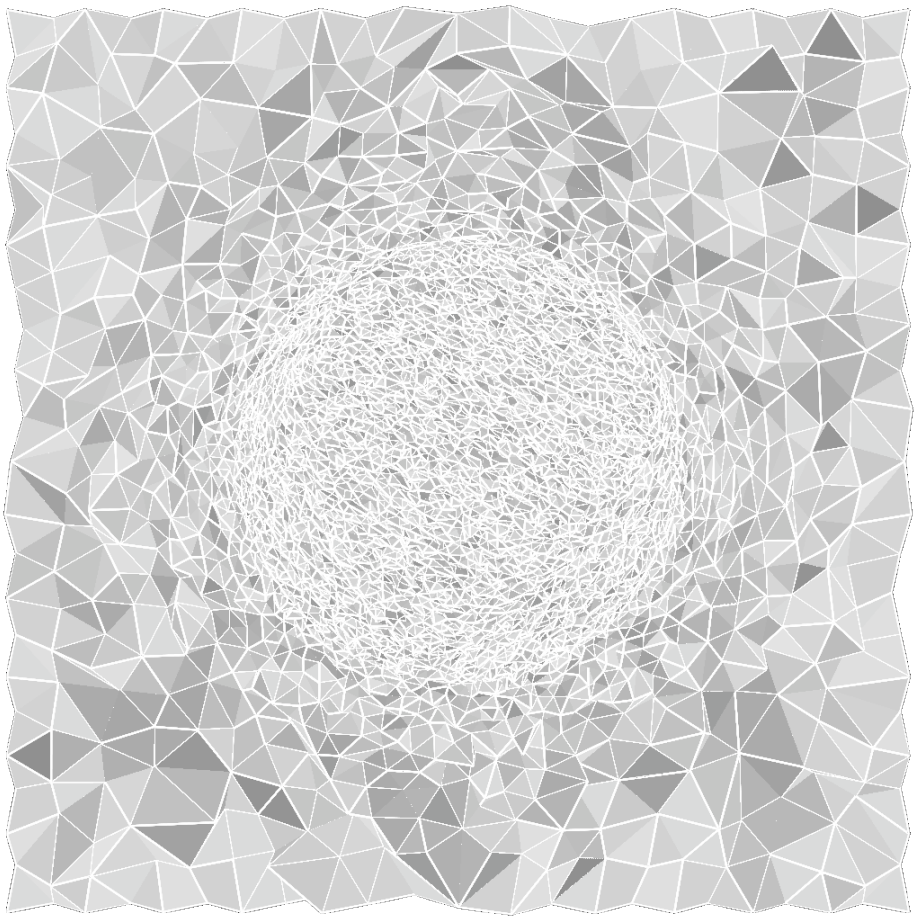

Fig. 22. Sphere volume grid for anisotropic case with direct source insertion and without transformation vector growth.

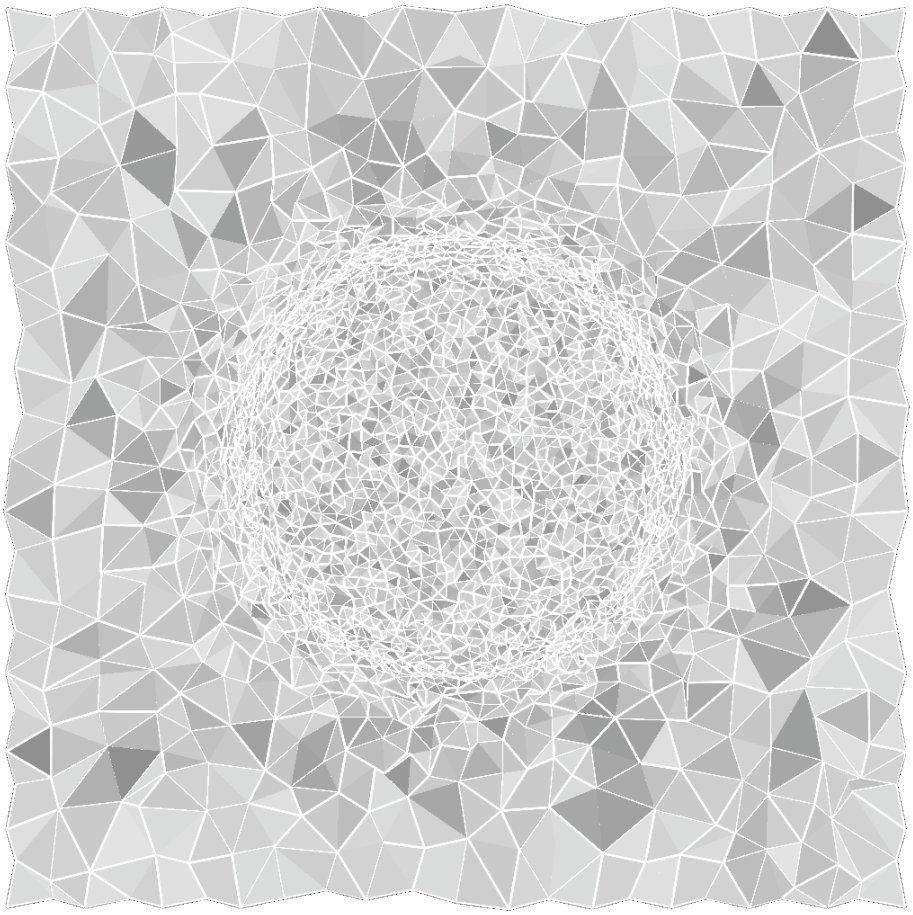

Fig. 23. Sphere volume grid for anisotropic case with direct source insertion and with transformation vector growth.

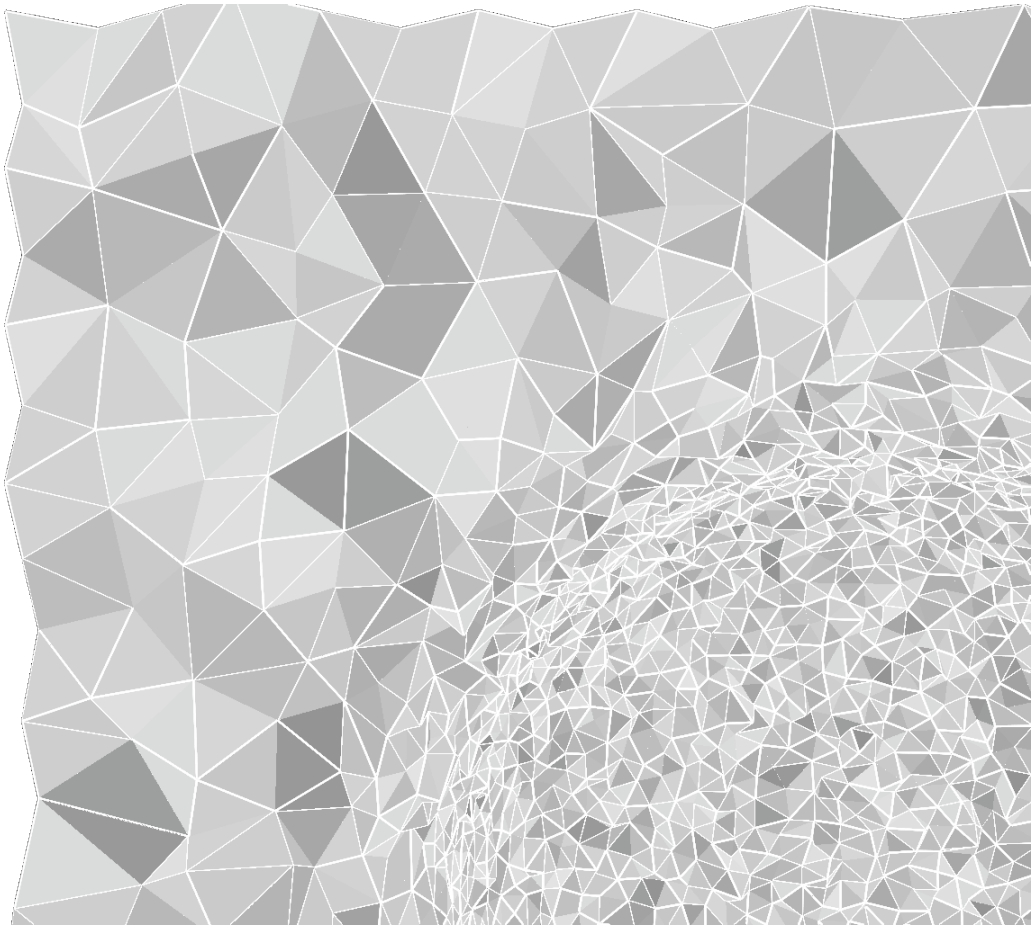

Fig. 24. Detail view of sphere volume grid for anisotropic case with direct source insertion and with transformation vector growth.

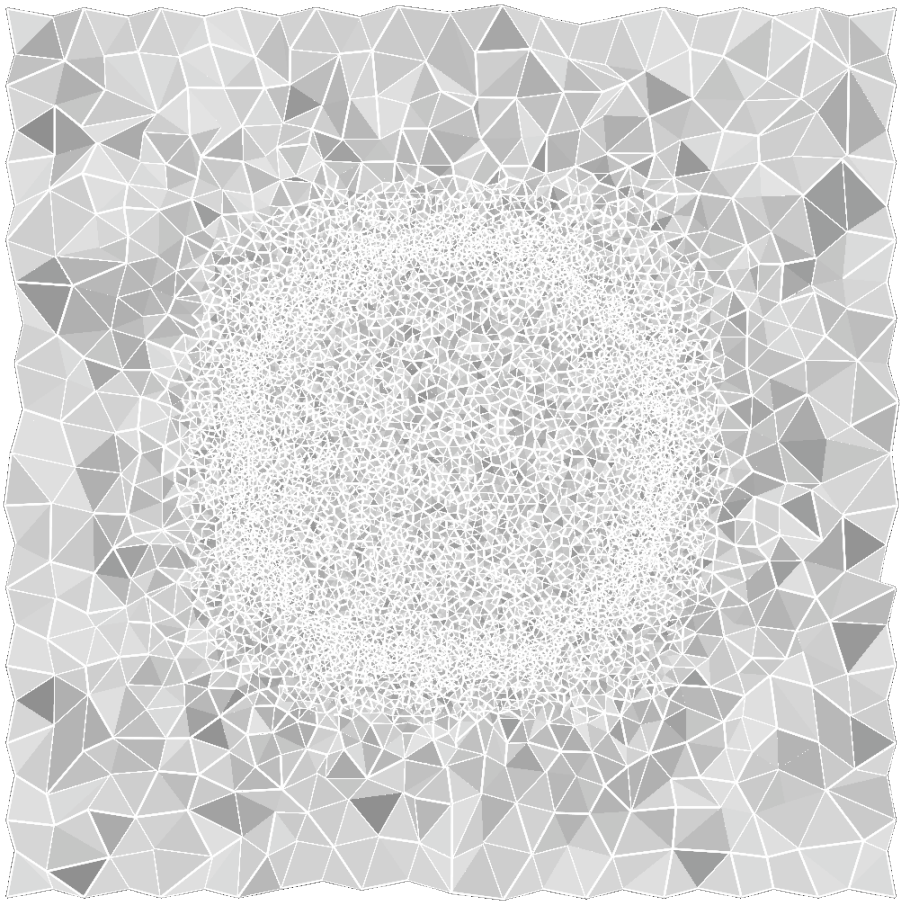

Fig. 25. Sphere volume grid for isotropic case with background source grid.

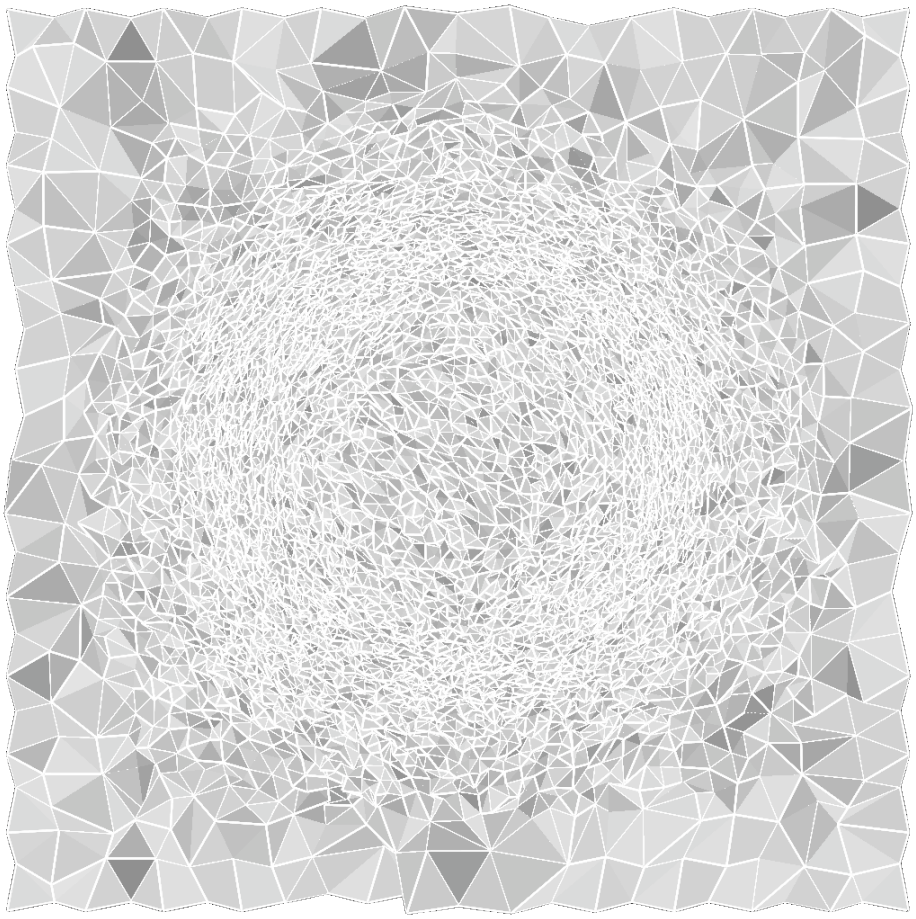

Fig. 26. Sphere volume grid for anisotropic case with background source grid and without transformation vector growth.

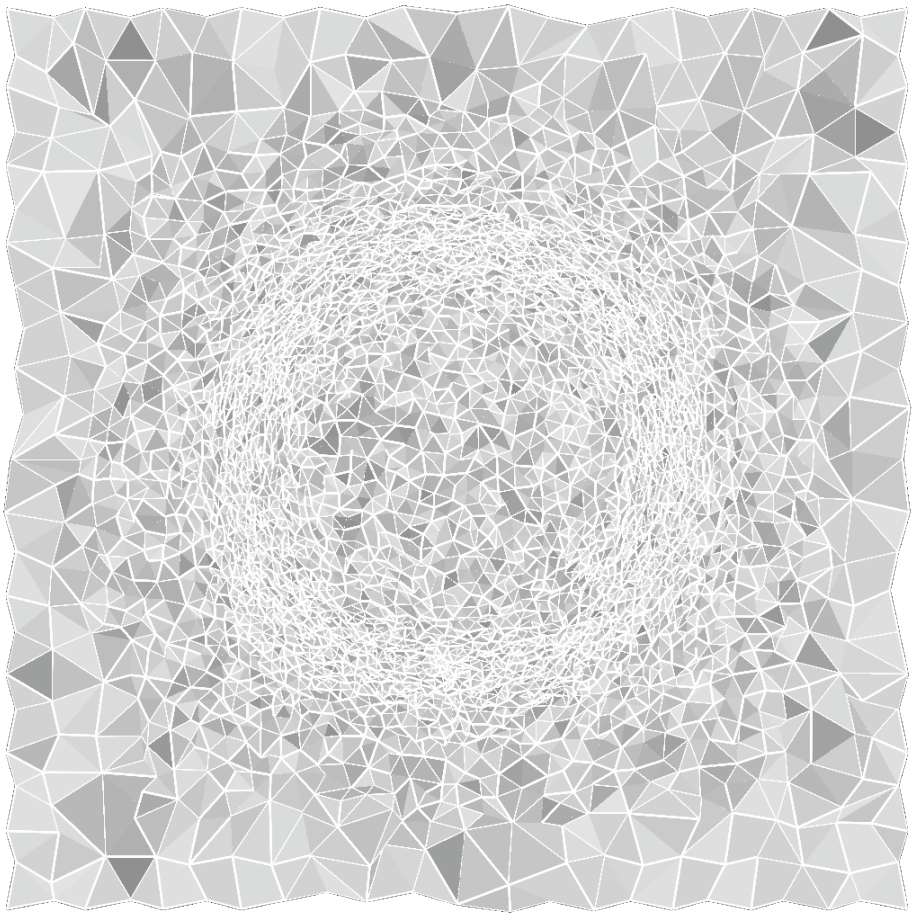

Fig. 27. Sphere volume grid for anisotropic case with background source grid and with transformation vector growth.

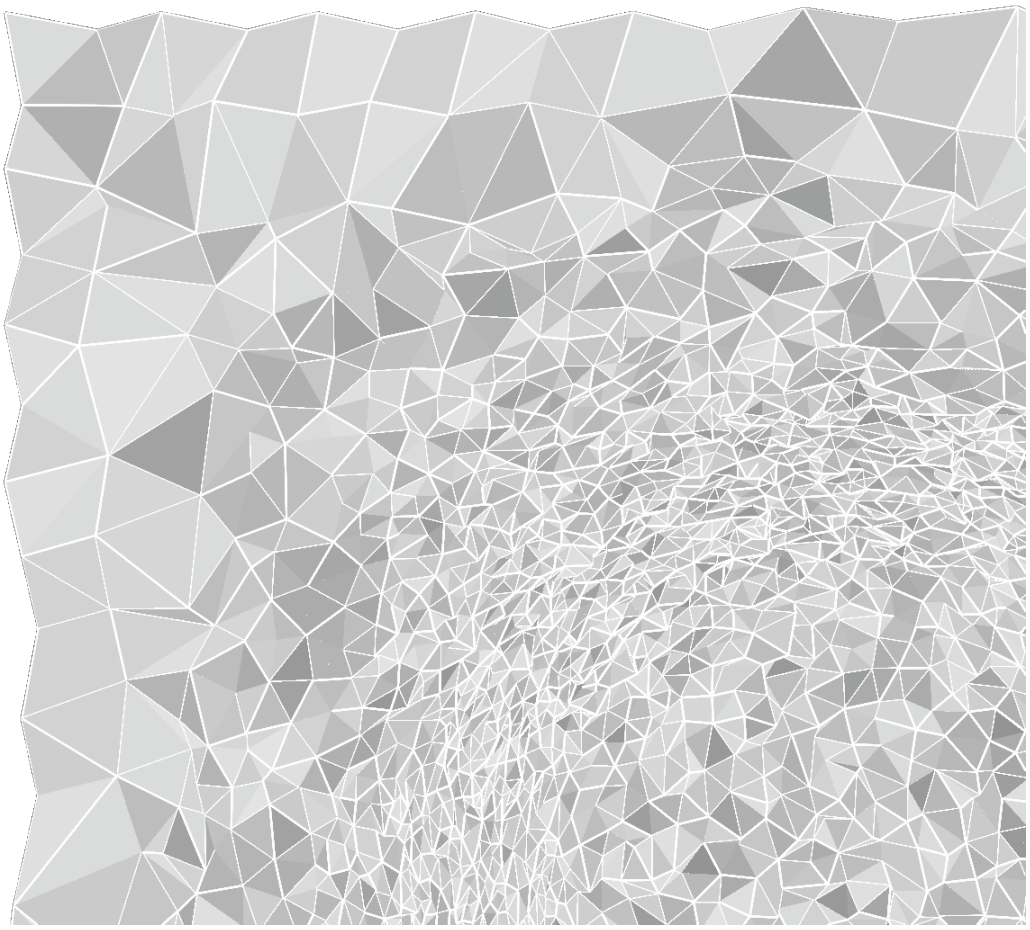

Fig. 28. Detail view of sphere volume grid for anisotropic case with background source grid and with transformation vector growth.

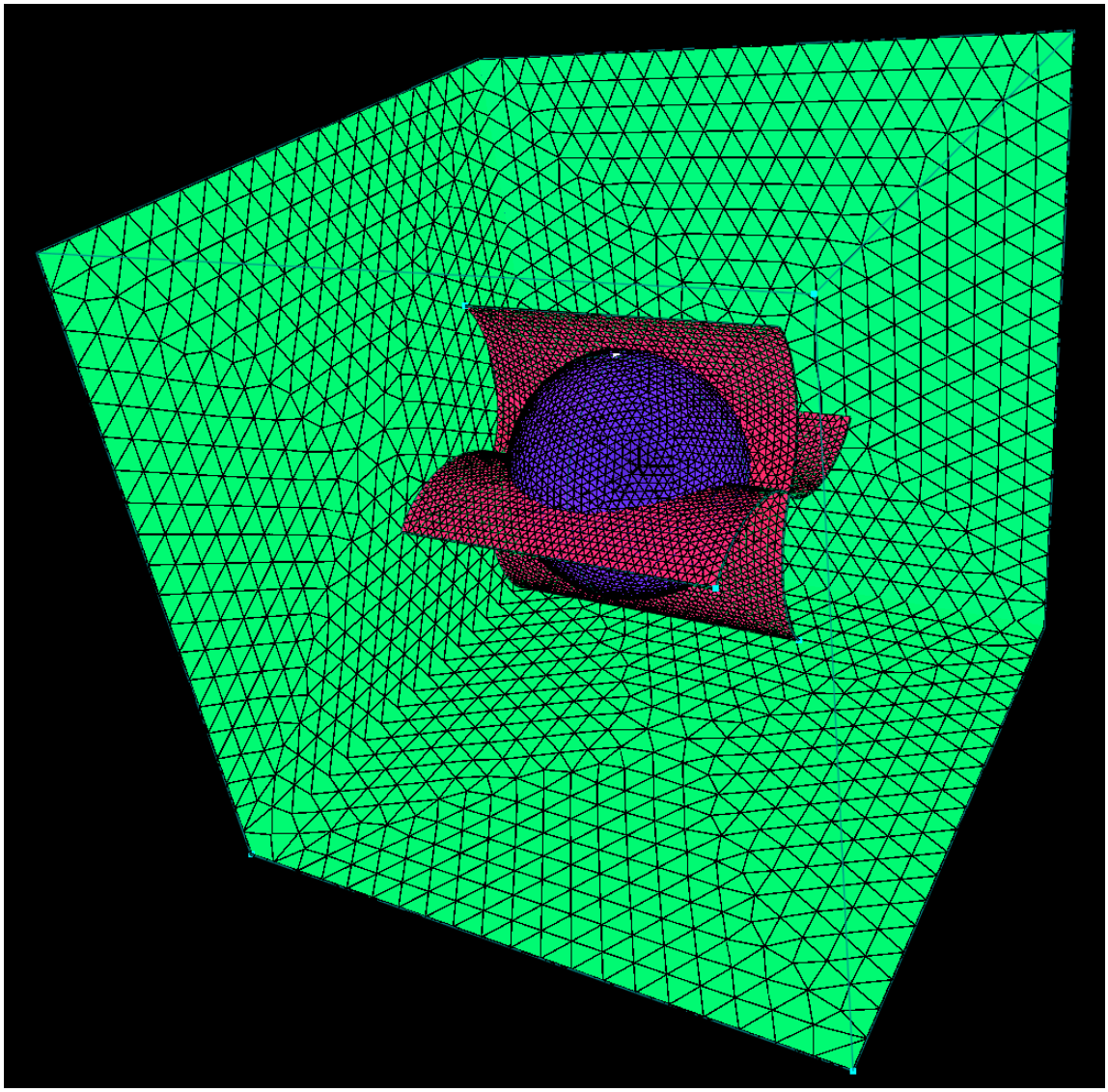

Fig. 29. Sphere & X-wave case surface grid.

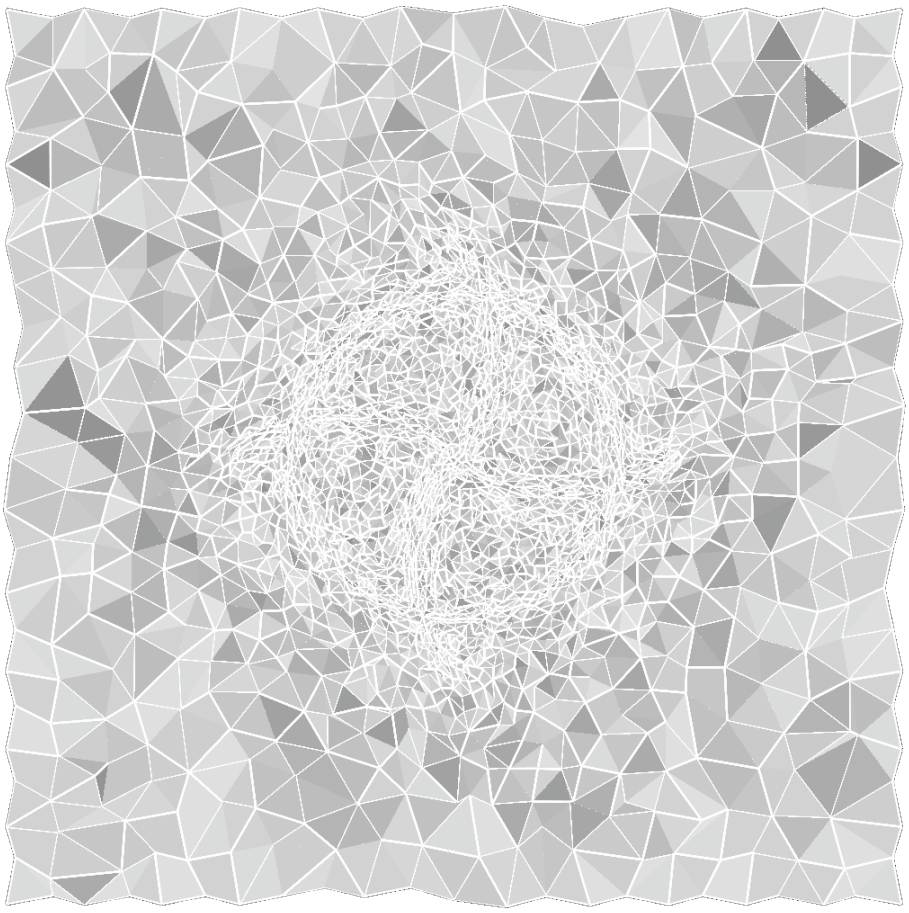

Fig. 30. Sphere & X-wave volume grid for anisotropic case with direct source insertion and with transformation vector growth.

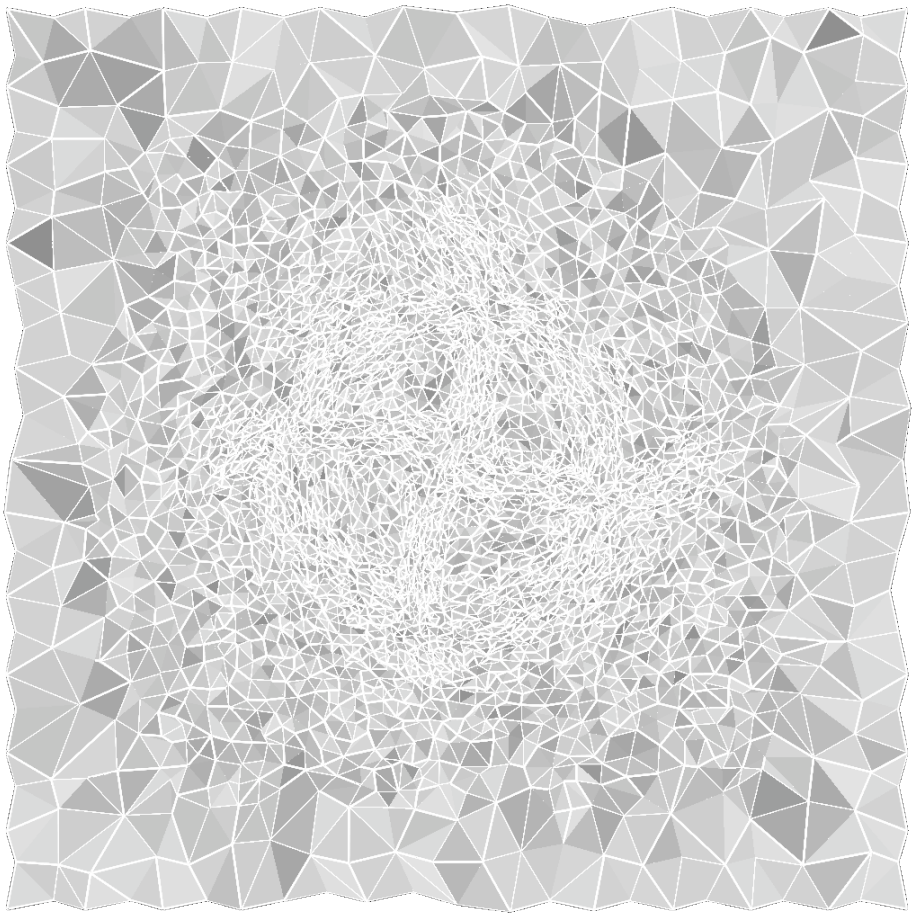

Fig. 31. Sphere & X-wave volume grid for anisotropic case with background source grid and with transformation vector growth.

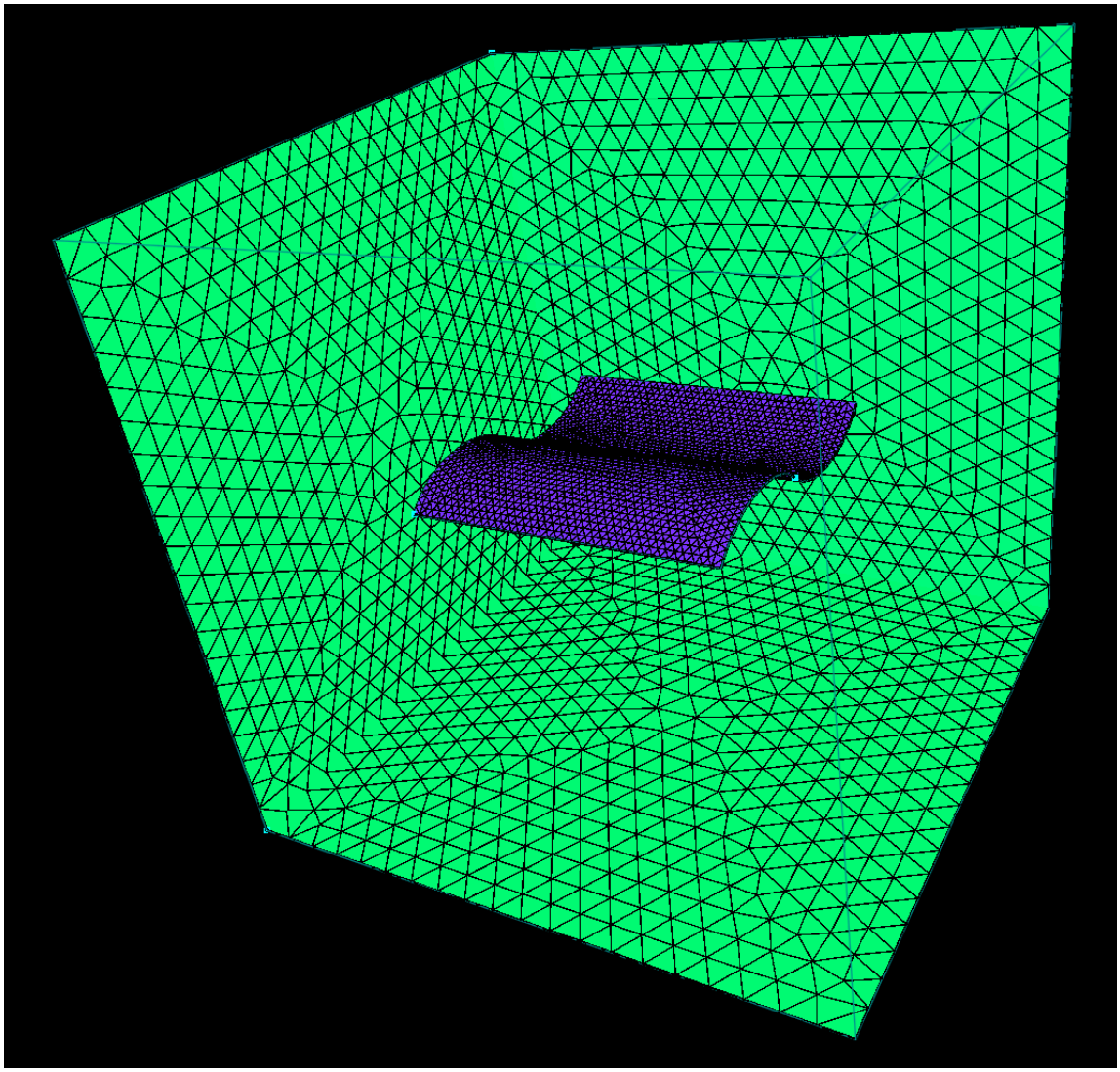

Fig. 32. Wave case surface grid.

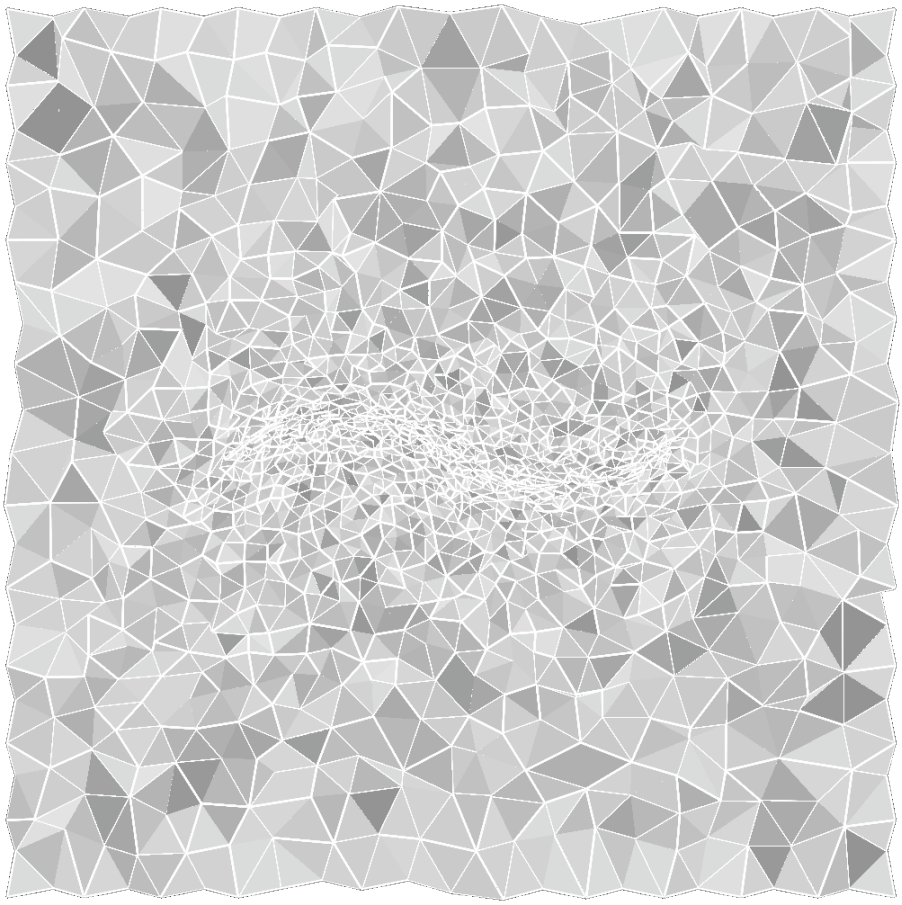

Fig. 33. Wave volume grid for anisotropic case with direct source insertion and with transformation vector growth.

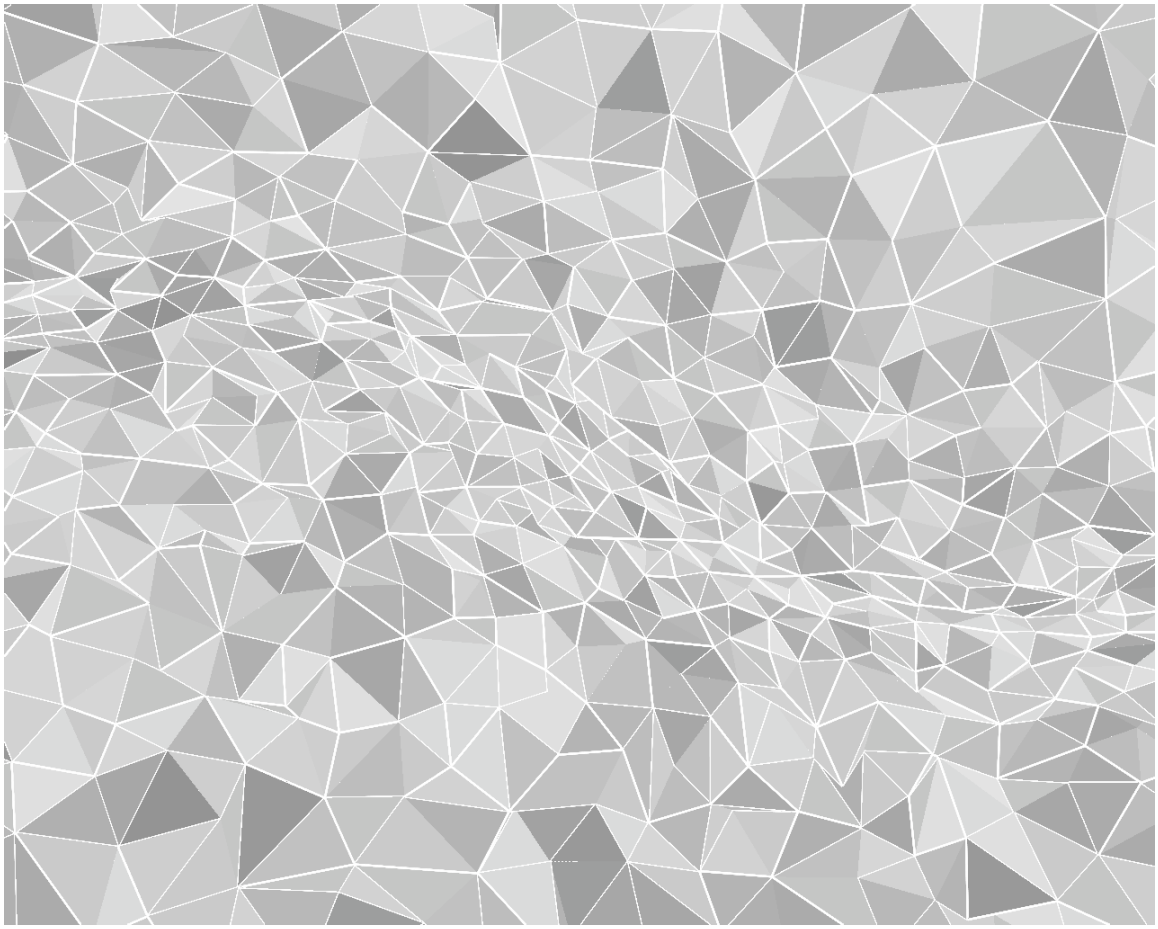

Fig. 34. Detail view of wave volume grid for anisotropic case with direct source insertion and with transformation vector growth.

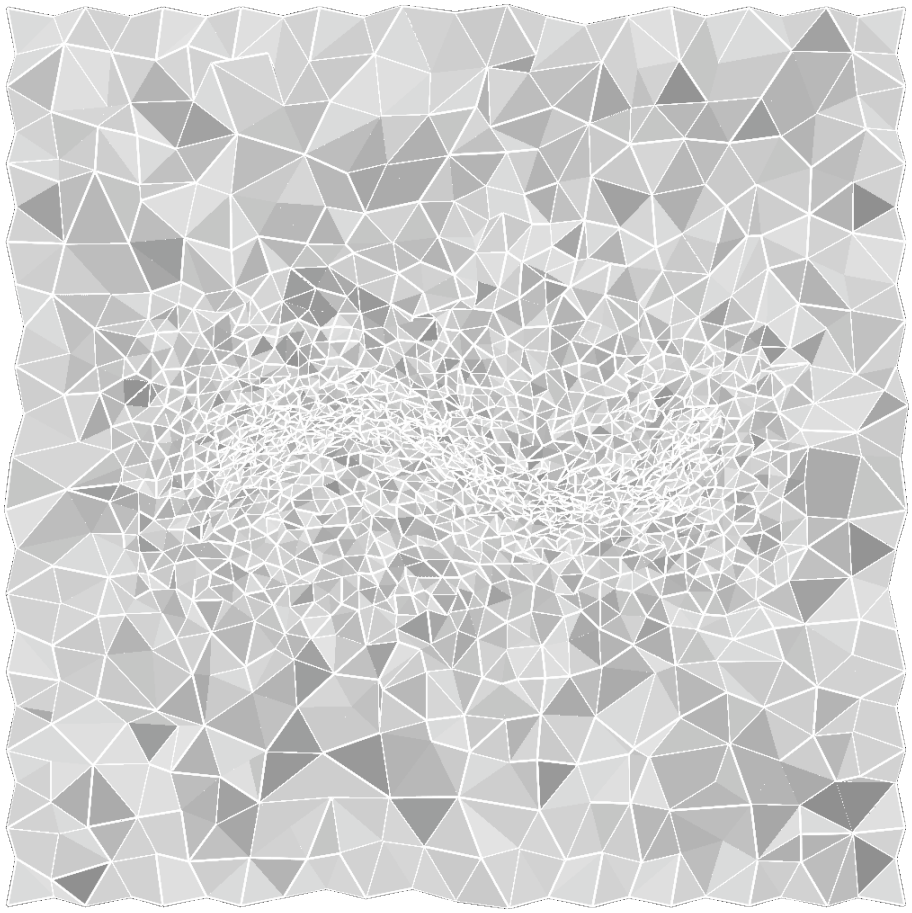

Fig. 35. Wave volume grid for anisotropic case with background source grid and with transformation vector growth.

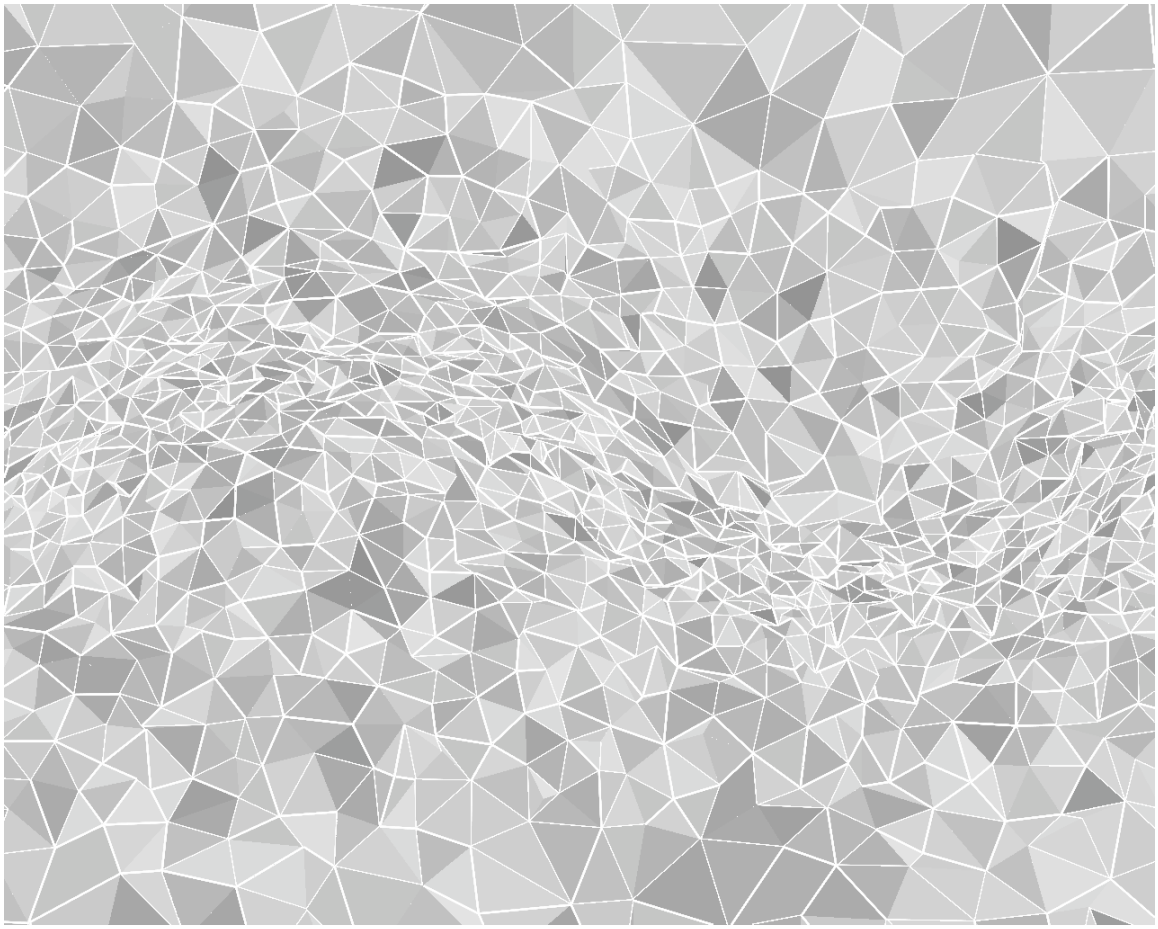

Fig. 36. Detail view of wave volume grid for anisotropic case with background source grid and with transformation vector growth.

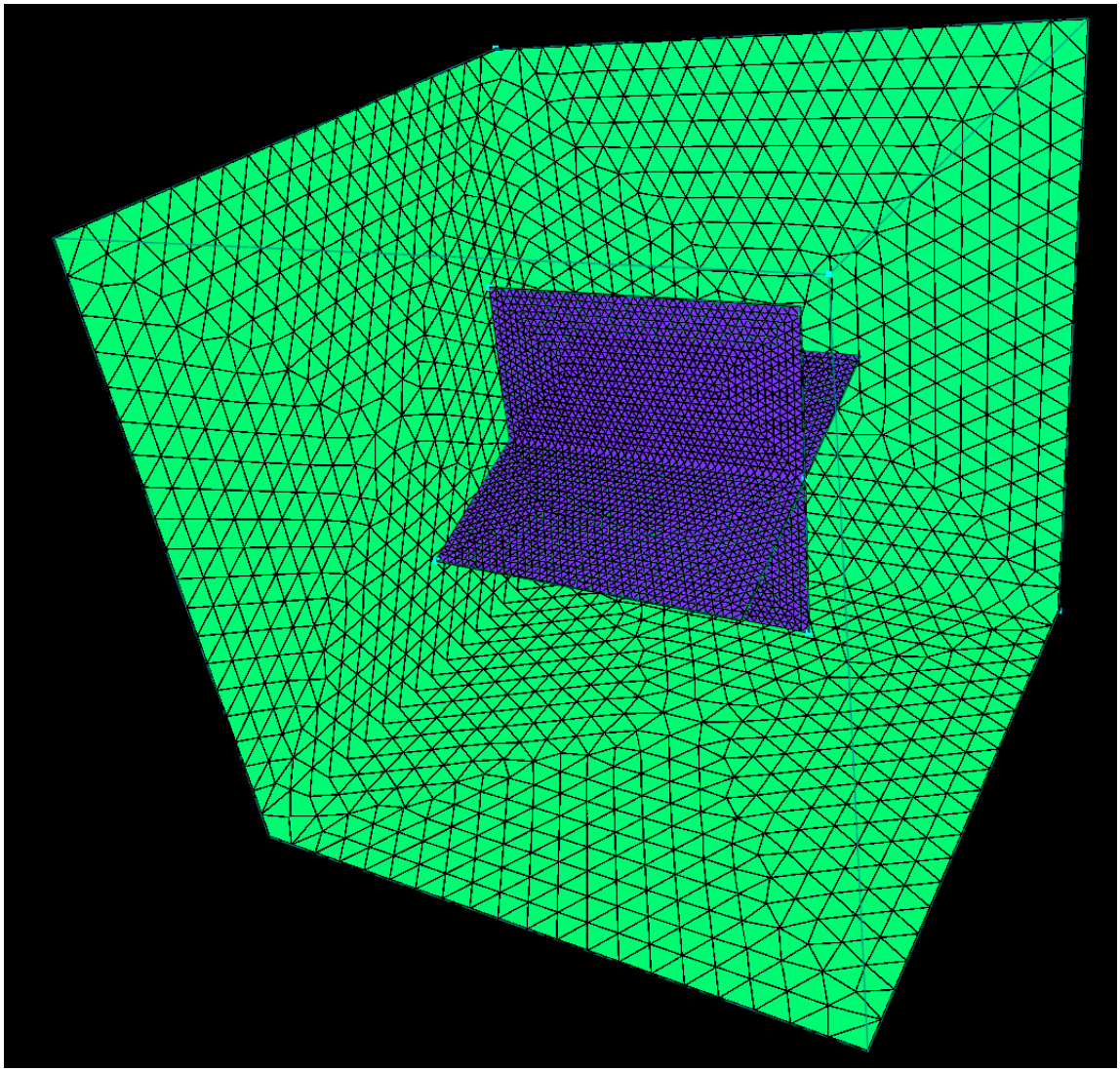

Fig. 37. X-plate case surface grid.

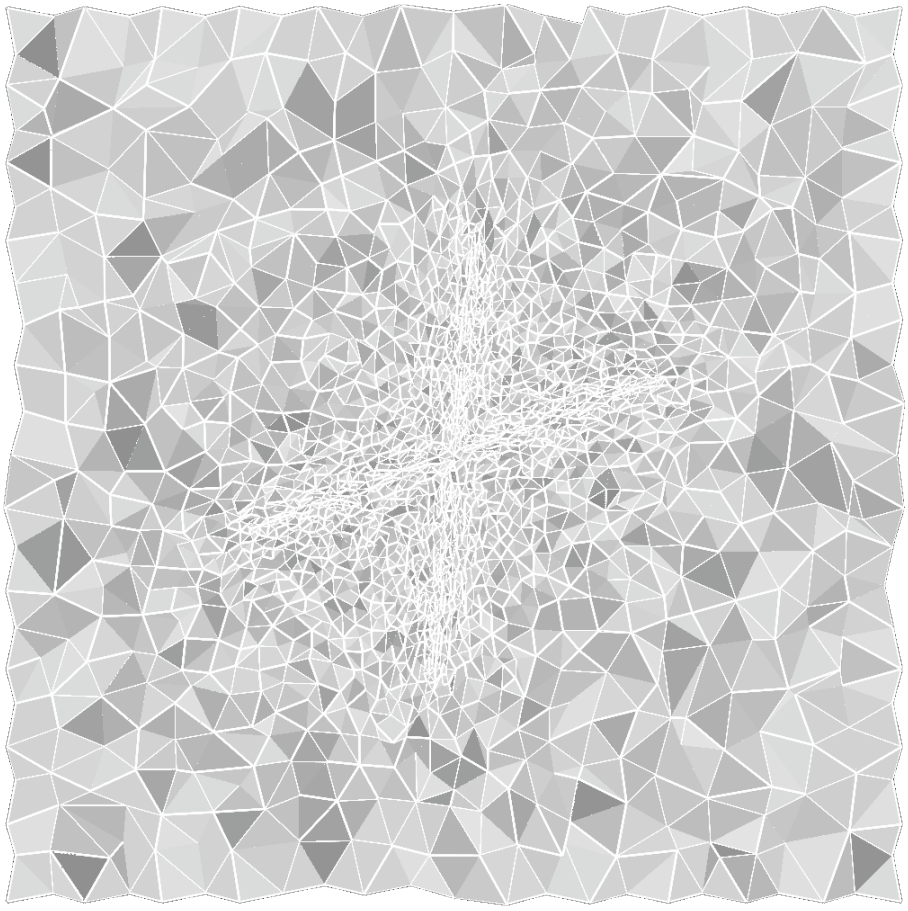

Fig. 38. X-plate volume grid for anisotropic case with direct source insertion and with transformation vector growth.

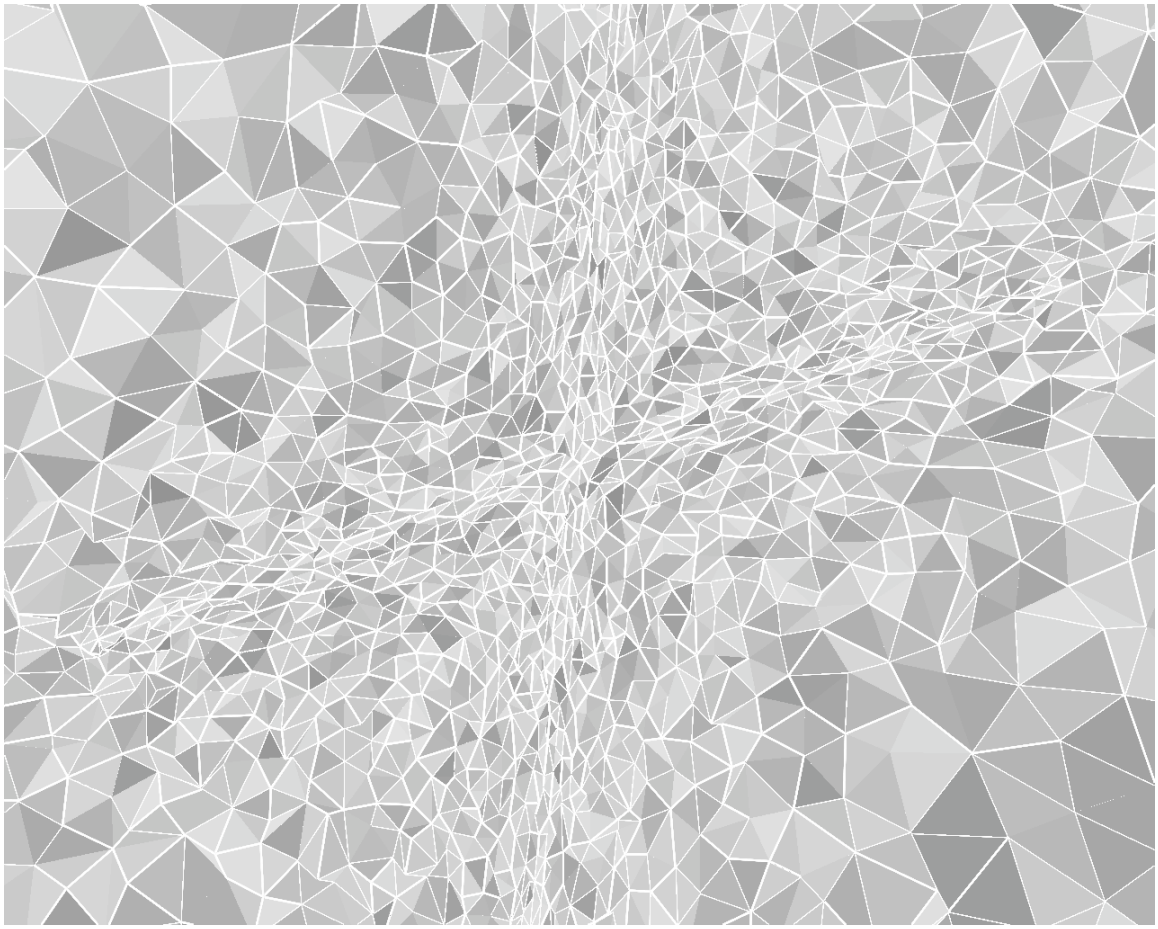

Fig. 39. Detail view of x-plate volume grid for anisotropic case with direct source insertion and with transformation vector growth.

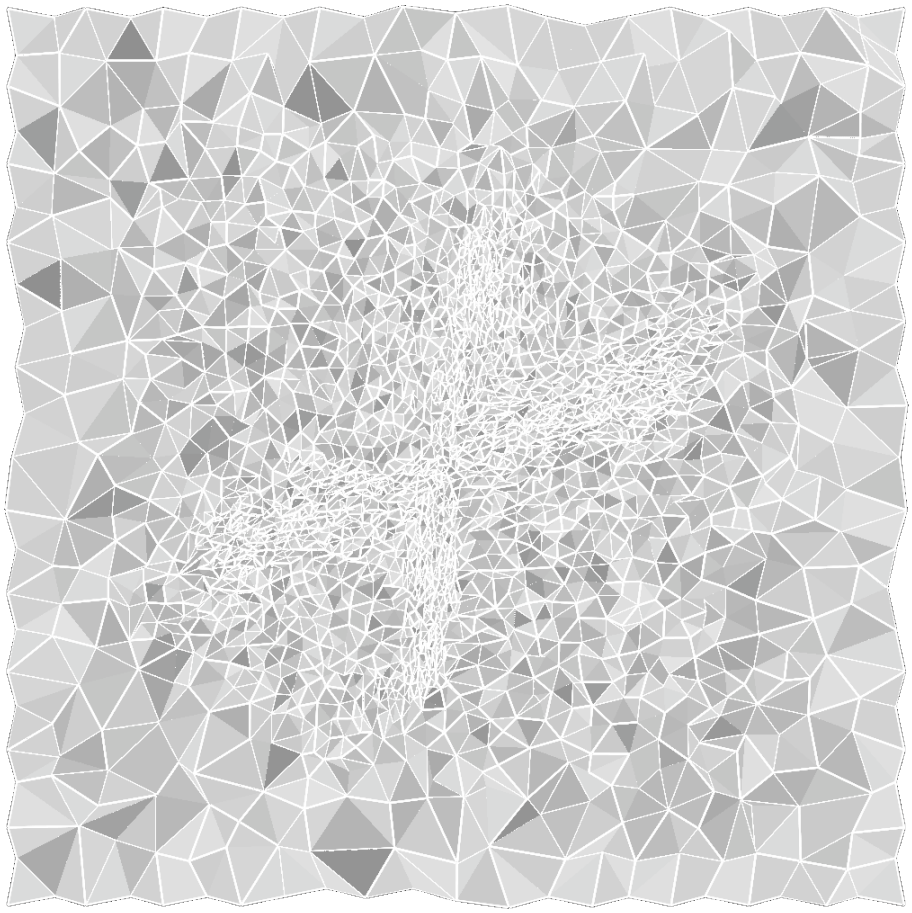

Fig. 40. X-plate volume grid for anisotropic case with background source grid and with transformation vector growth.

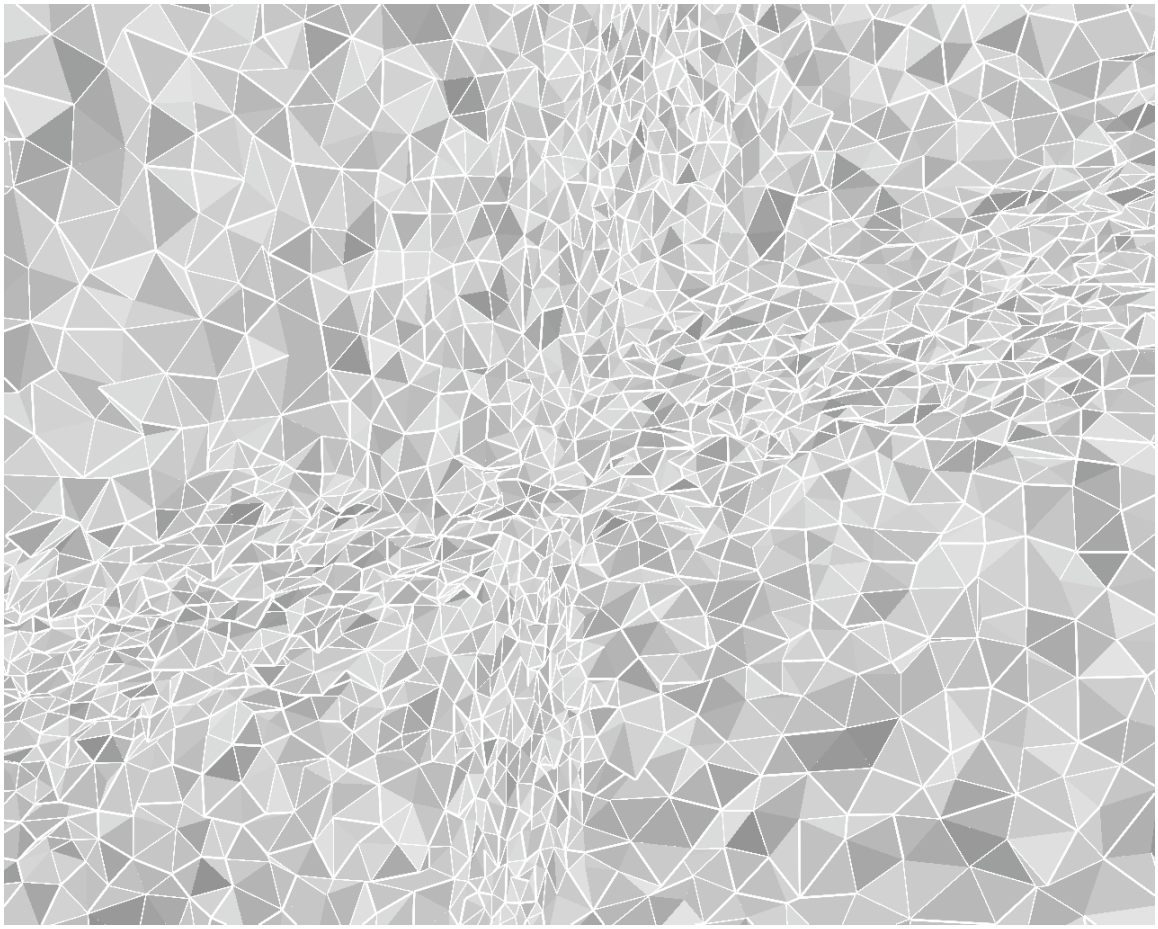

Fig. 41. Detail view of x-plate volume grid for anisotropic case with background source grid and with transformation vector growth.

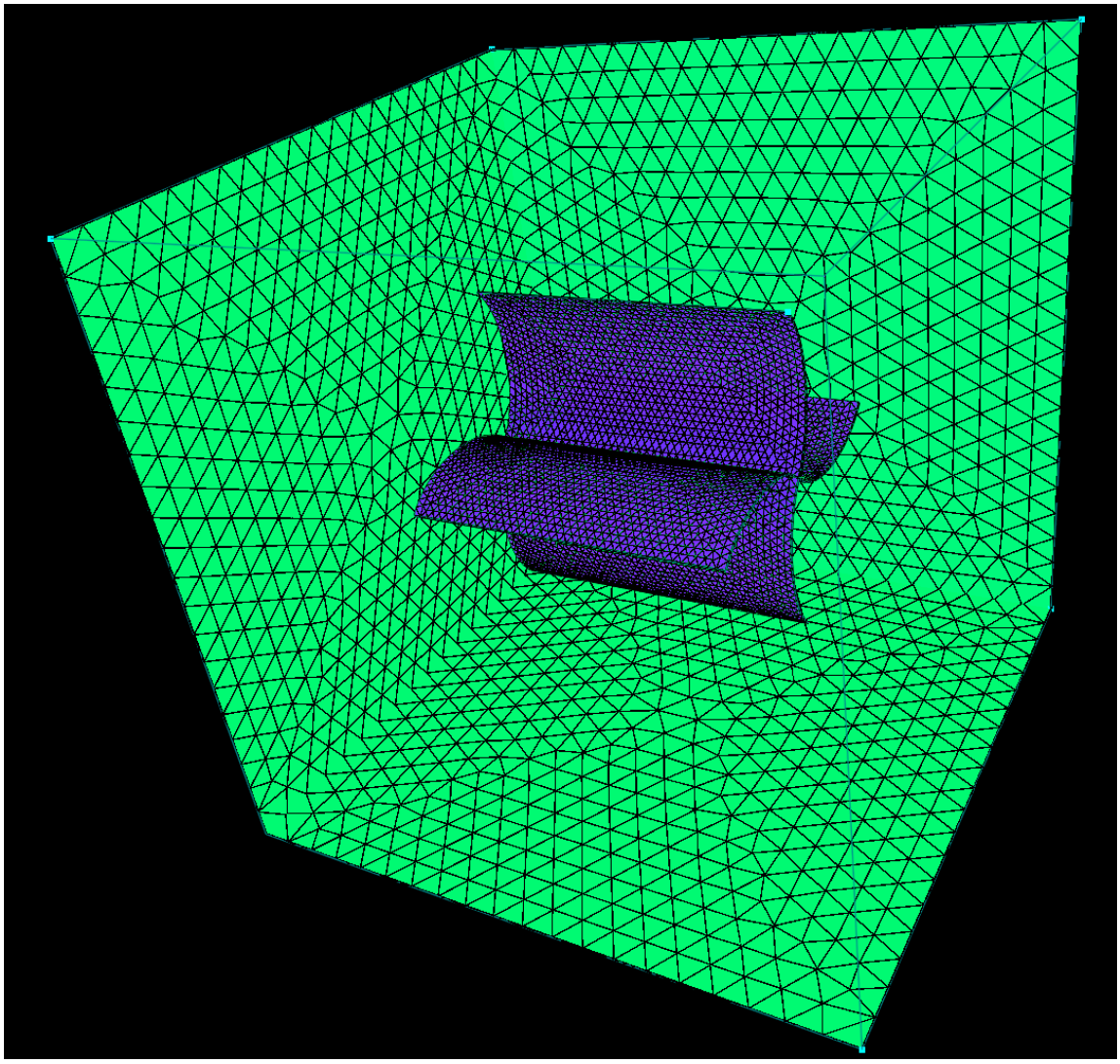

Fig. 42. X-wave case surface grid.

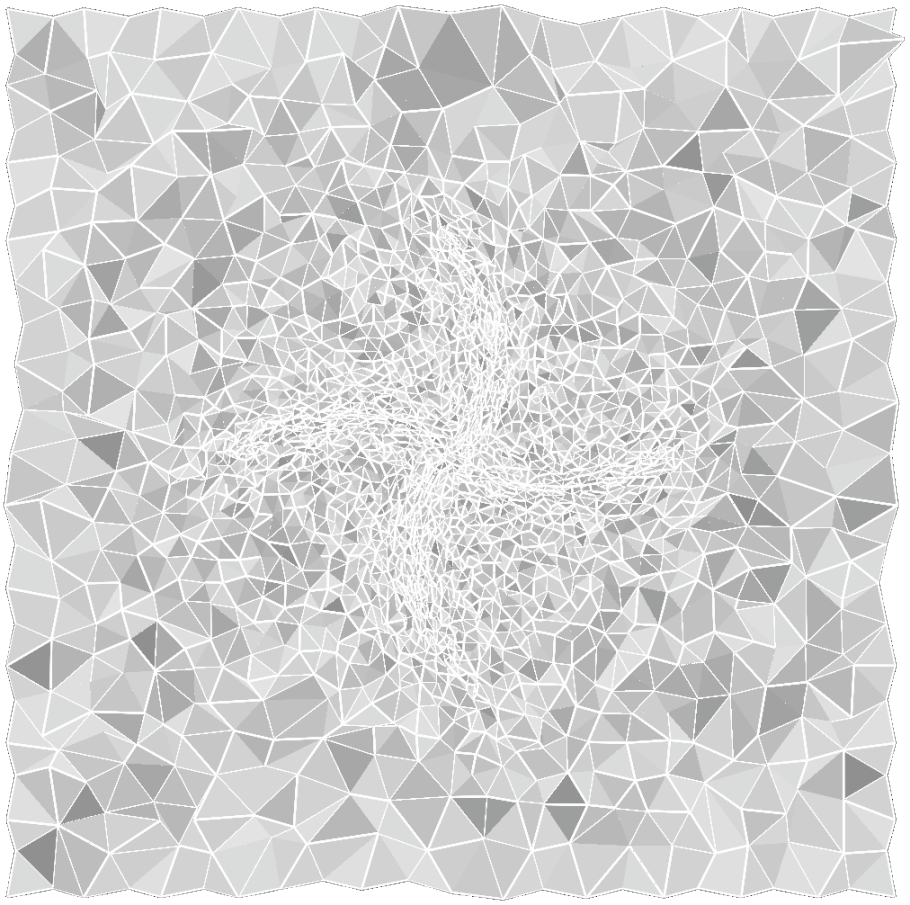

Fig. 43. X-wave volume grid for anisotropic case with direct source insertion and with transformation vector growth.

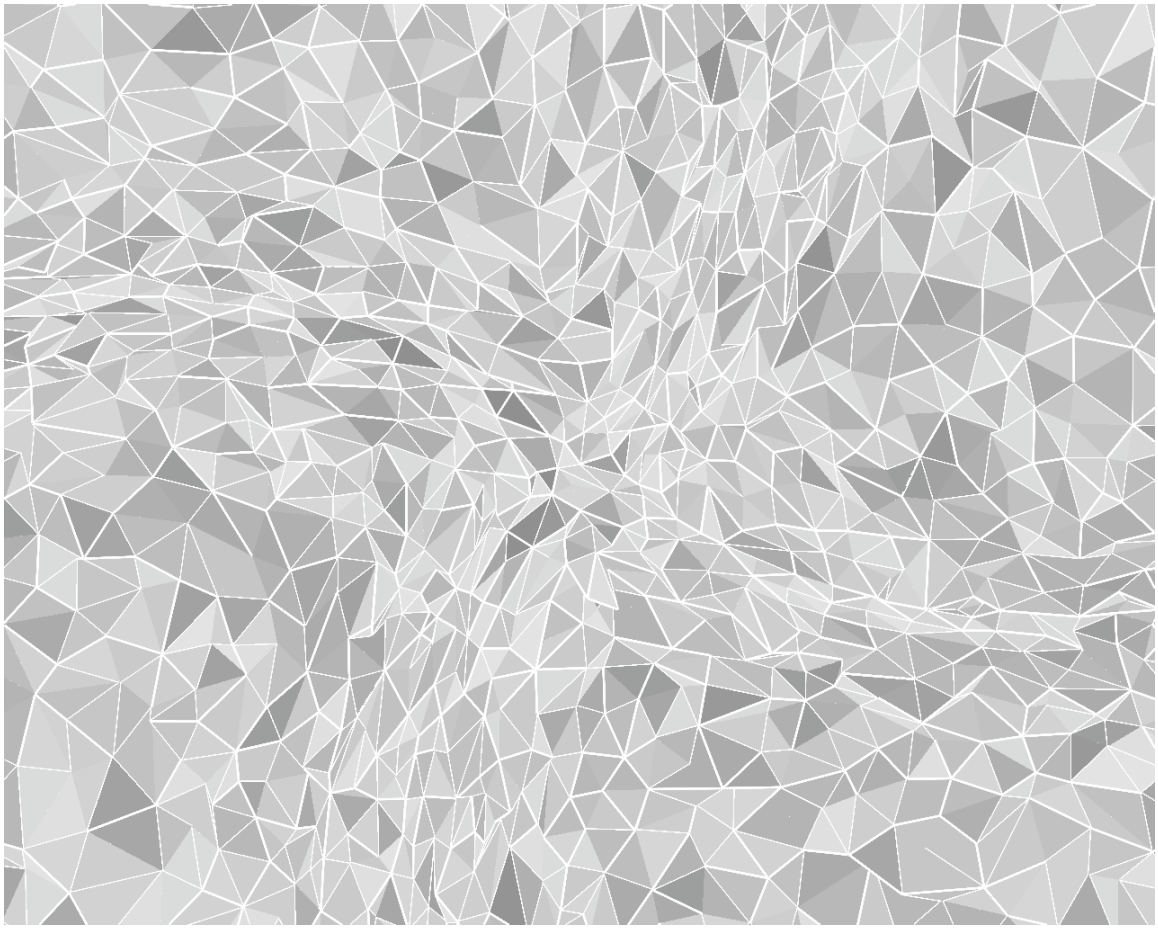

Fig. 44. Detail view of x-wave volume grid for anisotropic case with direct source insertion and with transformation vector growth.

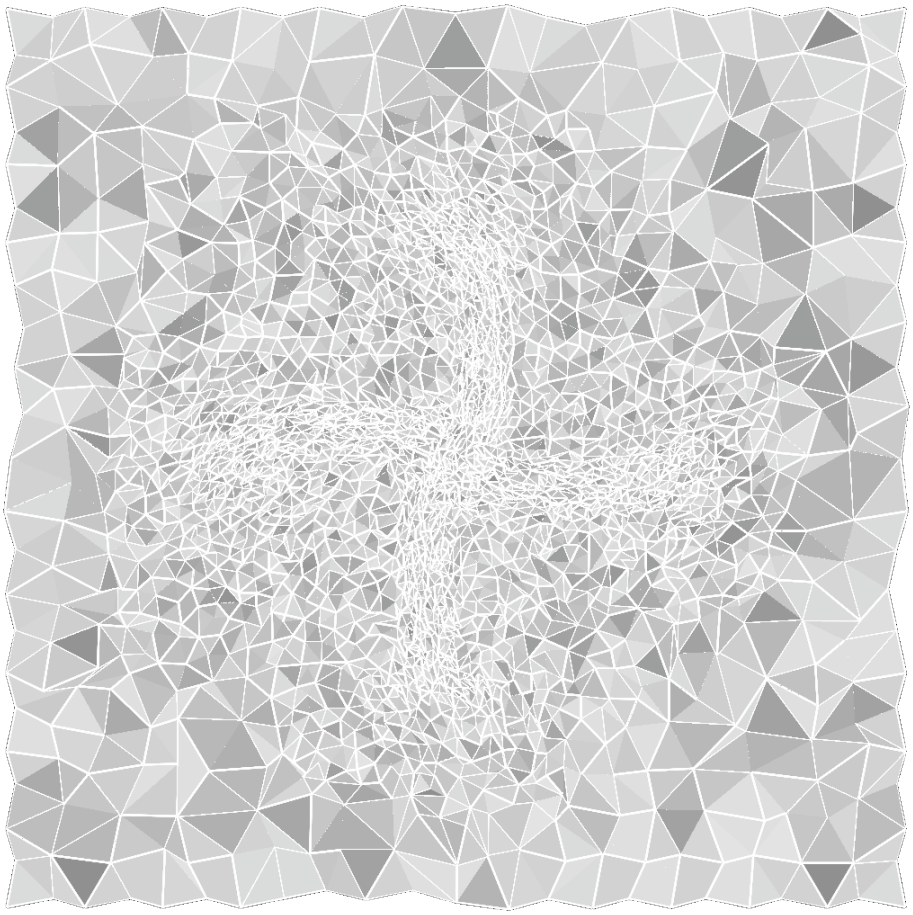

Fig. 45. X-wave volume grid for anisotropic case with background source grid and with transformation vector growth.

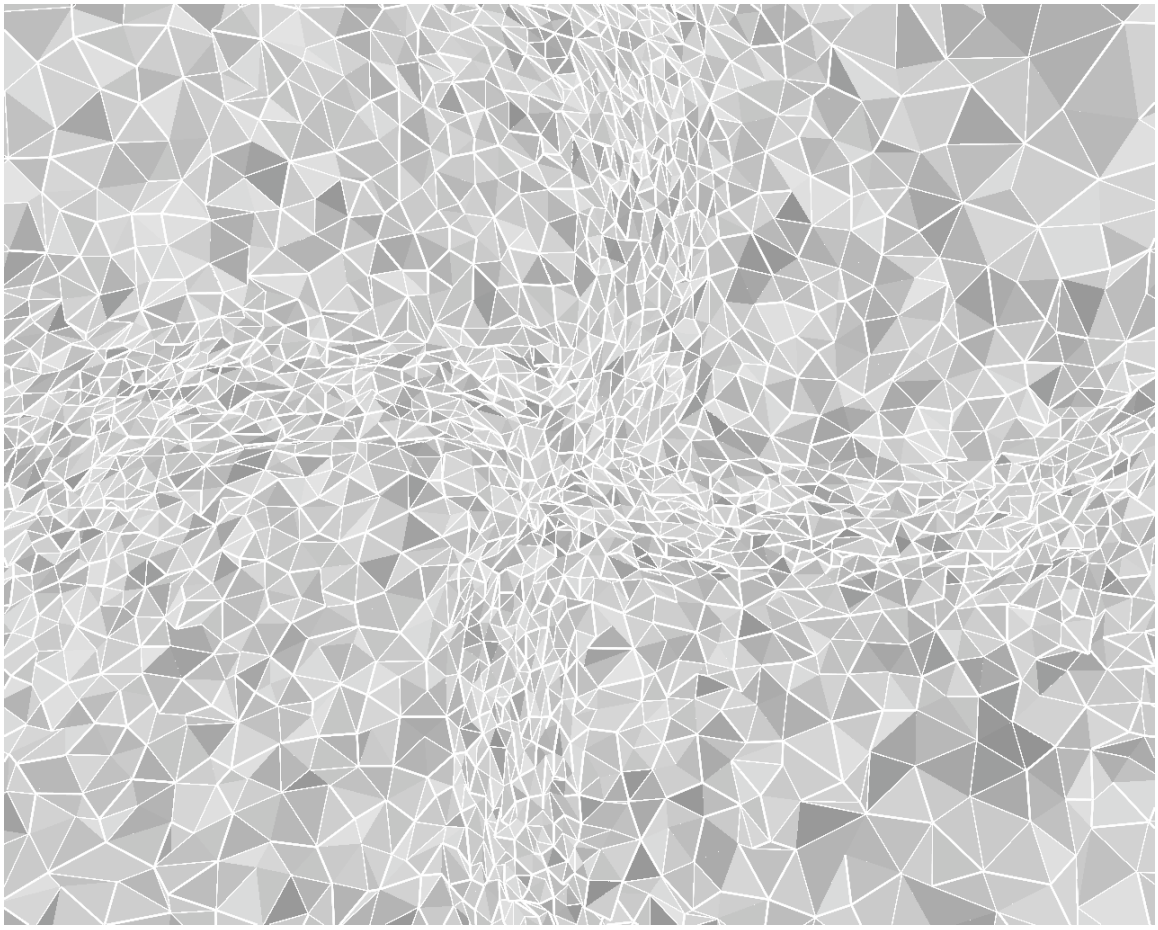

Fig. 46. Detail view of x-wave volume grid for anisotropic case with background source grid and with transformation vector growth.

4. Conclusion

An existing volume grid generation procedure, AFLR3, was successfully modified to generate anisotropic tetrahedral elements using a directional metric transformation defined at source nodes. The procedure can be coupled with a solver and an error estimator as part of an overall anisotropic solution adaptation methodology. It is suitable for use with an error estimator based on an adjoint, optimization, sensitivity derivative, or related approach. This offers many advantages, including more efficient point placement along with robust and efficient error estimation. It also serves as a framework for true grid optimization wherein error estimation and computational resources can be used as cost functions to determine the optimal point distribution.

Multiple two- and three-dimensional examples were presented that demonstrated the capability of the modified AFLR procedure to generate anisotropic elements using a set of source nodes with anisotropic transformation metrics. The example cases presented used moderate levels of anisotropy that resulted in usable element quality. Future testing with various flow solvers and methods for obtaining transformation metric information is needed to determine practical limits and evaluate the efficacy of the overall approach.

While extremely high transformation aspect ratios (>> 10:1) can be used with the modified AFLR3 procedure, the resulting grid quality degrades very quickly with increasing aspect ratio. Future work should explore the extension of the present approach coupled with the use of feature surfaces in select regions to better align the grid and maintain grid quality through more precise control of point placement. Feature surfaces can be treated as embedded boundaries and would provide a natural framework for directional refinement and automatic alignment with physical features. This is similar in approach to the generation of anisotropic elements within a boundary-layer region where the geometric boundary is used as a feature surface.